

Faculdade de Engenharia da Universidade do Porto



**Adaptação do modelo dinâmico de sistemas de
excitação recorrendo a uma meta-heurística do
tipo EPSO**

João Luis Rodrigues de Castro Mendes Martins

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de Computadores
Major Energia

Orientador: Prof. Dr. Helena Osório Pestana de Vasconcelos

19-07-2013

© João Luis Rodrigues de Castro Mendes Martins, 2013

Resumo

Neste trabalho foi desenvolvida uma ferramenta de apoio à realização de simulações dinâmicas e estudos de estabilidade de sistemas elétrico reais, através da realização do ajuste de parâmetros de um modelo dinâmico predefinido para sistema de excitação de máquinas síncronas, de modo a obter uma resposta no domínio das frequências e dos tempos semelhante à resposta de um outro modelo fornecido.

A metodologia utilizada consistiu na adaptação de uma meta-heurística do tipo EPSO (*Evolutionary Particle Swarm Optimization*) para resolver o problema. Este algoritmo foi testado para diferentes casos de estudo, tendo-se utilizado como modelo a ajustar o DC1A das normas IEEE (*Institute of Electrical and Electronics Engineers*).

Em todos os casos de estudo testados, os resultados obtidos confirmam a capacidade da metodologia desenvolvida em obter soluções de ajuste com qualidade.

Como resultado do trabalho, foi ainda desenvolvida uma nova técnica para impedir a convergência de meta-heurísticas do tipo EPSO e PSO (*Particle swarm optimization*) para ótimos locais. Concluiu-se também que o EPSO é ideal para resolver este tipo de problema uma vez que mostrou ser capaz de adaptar todas as funções de transferência testadas neste trabalho.

Abstract

In this work a tool was developed to aid dynamic simulations and stability studies in real power systems, by adjusting the parameter values of a dynamic model predefined for synchronous machine's excitation systems to obtain a time and frequency domain response similar to another supplied model.

The methodology applied consisted in the adaptation of an EPSO type meta-heuristics to solve the problem. This algorithm was tested for different study cases, having used as model to adjust, the DC1A model from the IEEE regulations.

In all tested study cases, the obtained results confirm the capacity to obtain adjust solutions with quality from the applied methodology.

As a result from this work, it was also developed a new technique to prevent the convergence of both EPSO and PSO meta-heuristics to local optima. It is also concluded that the EPSO is ideal to solve this kind of problems since it was capable of adapting all transfer functions tested in this work.

Agradecimentos

Este trabalho não seria o que é se não fosse pelo apoio da minha Orientadora Maria Helena Vasconcelos. Obrigado pela paciência, interesse e entusiasmo pelo meu trabalho.

Muitos agradecimentos à minha família, especialmente os meus pais Maria Gabriela e José Maria por sempre me terem apoiado quando precisei.

谢谢燕炜无尽的帮助，乐观的态度和无私的关爱。如果没有你，我不能以这种方式顺利完成这项工作。

Índice

Resumo	iii
Abstract.....	v
Agradecimentos	vii
Índice.....	ix
Lista de figuras	xi
Lista de tabelas	xiii
Abreviaturas e Símbolos	xv
Capítulo 1	16
Introdução.....	16
1.1 - Contexto e justificação do trabalho.....	16
1.2 - Objectivos.....	17
1.3 - Descrição geral dos restantes capítulos da tese.....	18
Capítulo 2	19
Ajuste dos parâmetros para modelos de sistemas de excitação.....	19
2.1 - Sistemas de excitação	19
2.2 - Sobre ajuste de parâmetros para modelos de sistemas de excitação	23
Capítulo 3	26
EPSO 26	
3.1 - Introdução.....	26
3.2 - Breve descrição de meta-heurísticas do tipo EC	26
3.3 - Breve descrição de meta-heurísticas do tipo PSO.....	27
3.4 - Breve descrição de meta-heurísticas do tipo EPSO	28
Capítulo 4	30
Metodologia desenvolvida	30
4.1 - Formulação do problema de otimização.....	30
4.2 - Implementação do EPSO.....	33
Capítulo 5	38

Casos de estudo	38
5.1 - Modelos fornecidos	38
5.2 - Casos de estudo do modelo 1	39
5.3 - Casos de estudo do modelo 2	49
5.4 - Adaptação do modelo 1 ao modelo 2 dos casos de estudo	60
Capítulo 6	63
Conclusões	63
Referências	64
Anexos	65

Lista de figuras

Figura 2.1 - Modelo do diagrama de blocos utilizado para este estudo, adaptado de [2]	20
Figura 2.2 - Diagrama de blocos do modelo DC1A, adaptado de [1].....	21
Figura 2.3 - Bloco simplificado que emula o regulador de tensão [1]	22
Figura 2.4 - Blocos que modelizam as funções de estabilização do regulador de tensão [1] ..	22
Figura 2.5 - Bloco simplificado que emula a excitatriz [1]	23
Figura 3.1 - Representação da equação do movimento sobre uma partícula	29
Figura 4.1 - Diagrama de blocos genérico de uma função de transferência com malha de realimentação	30
Figura 4.2 - Diagrama de blocos do modelo a ajustar DC1A e do modelo fornecido como caso base em simulink.....	31
Figura 4.3 - Representação do ajuste da resposta $g(s,x)$ a $h(s)$ com s no domínio C utilizando a aplicação desenvolvida nesta dissertação, com a função objetivo salientada.....	32
Figura 4.4 - Diagrama de Bode do módulo da resposta do ótimo global obtido (DC1A) e do modelo fornecido.....	35
Figura 4.5 - Diagrama de Bode da fase da resposta do ótimo global obtido (DC1A) e do modelo fornecido.....	35
Figura 4.6 - Gráfico tridimensional da resposta do ótimo global obtido (DC1A) e do modelo fornecido.....	36
Figura 4.7 - Soma das distâncias das partículas ao ponto médio do enxame em cada iteração	37
Figura 4.8 - Evolução do valor da função objetivo do ótimo global ao longo das iterações	37
Figura 5.1 - Esquema de simulink utilizado para simular o primeiro modelo dos casos de estudo	38
Figura 5.2 - Esquema simulink utilizado para validar o ajuste obtido para o segundo modelo fornecido.....	39

Figura 5.3 - Diagramas de bode - Módulo (solução obtida para o caso base)	41
Figura 5.4 - Diagramas de bode - Fase (solução obtida para o caso base).....	41
Figura 5.5 - Resposta temporal de V (solução obtida para o caso base)	42
Figura 5.6 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 1)	43
Figura 5.7 - Diagramas de bode - Fase (solução obtida para o caso de estudo 1).....	44
Figura 5.8 - Resposta temporal de V (solução obtida para o caso de estudo 1)	44
Figura 5.9 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 2)	45
Figura 5.10 - Diagramas de bode - Fase (solução obtida para o caso de estudo 2)	46
Figura 5.11 - Resposta temporal de V (solução obtida para o caso de estudo 2).....	46
Figura 5.12 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 3)	47
Figura 5.13 - Diagramas de bode - Fase (solução obtida para o caso de estudo 3)	48
Figura 5.14 - Resposta temporal de V (solução obtida para o caso de estudo 3).....	48
Figura 5.15 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 4)	50
Figura 5.16 - Diagramas de bode - Fase (solução obtida para o caso de estudo 4)	50
Figura 5.17 - Resposta temporal de V (solução obtida para o caso de estudo 4).....	51
Figura 5.18 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 5)	52
Figura 5.19 - Diagramas de bode - Fase (solução obtida para o caso de estudo 5)	53
Figura 5.20 - Resposta temporal de V (solução obtida para o caso de estudo 5).....	53
Figura 5.21 - Diagramas de bode - Módulo (solução obtida para a nova solução do caso de estudo 5).....	54
Figura 5.22 - Diagramas de bode - Fase (solução obtida para a nova solução do caso de estudo 5).....	55
Figura 5.23 - Resposta temporal de V (solução obtida para a nova solução do caso de estudo 5).....	55
Figura 5.24 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 6)	56
Figura 5.25 - Diagramas de bode - Fase (solução obtida para o caso de estudo 6)	57
Figura 5.26 - Resposta temporal de V (solução obtida para o caso de estudo 6).....	57
Figura 5.27 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 7)	58
Figura 5.28 - Diagramas de bode - Fase (solução obtida para o caso de estudo 7)	59
Figura 5.29 - Resposta temporal de V (solução obtida para o caso de estudo 7).....	59
Figura 5.30 - Diagramas de bode - Módulo (solução obtida para a adaptação do modelo fornecido 1 ao caso de estudo 4)	60

Figura 5.31 - Diagramas de bode - Fase (solução obtida para a adaptação do modelo fornecido 1 ao caso de estudo 4)	61
Figura 5.32 - Gráfico tridimensional da adaptação do modelo 1 ao caso de estudo 4	62
Figura 5.33 - Resposta temporal de V (solução obtida para o caso de estudo 7).....	62

Lista de tabelas

Tabela 5.1 - Casos de estudo do modelo 1	40
Tabela 5.2 - Valores dos parâmetros DC1A obtidos para o caso base	40
Tabela 5.3 - Valores dos parâmetros para o modelo DC1A obtidos para o caso de estudo 1 ...	42
Tabela 5.4 - Valores dos parâmetros para o modelo DC1A obtidos para o caso de estudo 2 ...	45
Tabela 5.5 - Valor dos parâmetros para o modelo DC1A obtidos para o caso de estudo 3	47
Tabela 5.6 - Casos de estudo do modelo 2	49
Tabela 5.7 - Valor dos parâmetros do modelo DC1A obtidos para o caso de estudo 4	49
Tabela 5.8 - Valor dos parâmetros do modelo DC1A obtidos para o caso de estudo 5	51
Tabela 5.9 - Valor dos parâmetros do modelo DC1A obtidos para o caso de estudo 6	56
Tabela 5.10 - Valor dos parâmetros do modelo DC1A obtidos para o caso de estudo 7	58
Tabela 5.11 - Parâmetros obtidos para o ajuste do modelo fornecido 1	60

Abreviaturas e Símbolos

Lista de abreviaturas (ordenadas por ordem alfabética)

AVR	Regulador Automático de Tensão (<i>Automatic Voltage Regulator</i>)
EC	<i>Evolutionary computation</i>
EPSO	<i>Evolutionary Particle Swarm Optimization</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
PSO	<i>Particle swarm optimization</i>
SEE	Sistema elétrico de energia

Lista de símbolos

ω	Frequência angular
σ	Componente real de um número complexo
j	Número imaginário
s	Número complexo $s = \sigma + j \omega$

Capítulo 1

Introdução

Neste trabalho foi desenvolvido um algoritmo que modifica o valor dos parâmetros de uma função complexa de forma a aproximar a sua resposta a outra função com resposta fixa num determinado domínio das frequências, através de uma meta-heurística do tipo EPSO. O domínio da aplicação consistiu em aplicar um modelo dinâmico *standard* do sistema de excitação de uma máquina síncrona convencional, de modo a que este se adapte à resposta de um outro modelo dinâmico fornecido para o mesmo sistema de excitação. Para os casos de estudo analisados, como modelo *standard* foi utilizado o modelo IEEE DC1A [1], tendo sido testada a sua capacidade de se adaptar a diferentes alternativas para o modelo fornecido. O domínio considerado para a resposta do sistema foi para frequências entre os 0 e 3 Hz, pois segundo as normas IEEE [1], este é o domínio de frequências para o qual os modelos *standard* de sistemas de excitação são válidos.

1.1 - Contexto e justificação do trabalho

Com o aumento da dimensão dos sistemas elétricos de energia, surge a necessidade de uma análise mais detalhada das perturbações credíveis que o sistema elétrico de energia pode suportar sem que ocorram violações dos limites de operacionalidade, nem interrupção de serviço. Para efetuar esta análise, é necessário realizar a simulação dinâmica do sistema, uma ferramenta matemática que calcula a resposta transitória do sistema, para uma dada perturbação. Para isso, é necessário um modelo do sistema e para cada componente foram criados modelos matemáticos *standard* que são utilizados depois nos programas de simulação dinâmica. Num sistema elétrico real, nem sempre os modelos fornecidos pelo fabricante correspondem aos modelos *standard*, pelo que, é necessário um programa que a partir da resposta no domínio das frequências do modelo fornecido pelo fabricante, ou pura e

simplesmente pela resposta dinâmica obtida da simulação dinâmica de perturbações, consiga ajustar um modelo standard, minimizando a diferença entre as respostas.

Quase todos os problemas existentes possuem diversas formas de resolução, sendo um exemplo notável o caso da comunicação: diferentes seres humanos, diferentes condições e locais foram gerando inúmeros dialetos. Quando o número de formas de solução se torna elevado, torna-se interessante o aparecimento de agentes capazes de compatibilizar as várias formas de solução por forma a utilizar o melhor de cada. No caso da comunicação, esses agentes são os tradutores, ou por exemplo no caso da compatibilização das linguagens de programação com linguagem máquina, os compiladores.

O algoritmo desenvolvido surge também, como um desses agentes, para dar resposta aos inúmeros modelos para sistemas de excitação (as formas de resolução) e compatibiliza-os ajustando a resposta de um modelo para ser semelhante a outro num determinado domínio. Mais concretamente, no desafio proposto pela dissertação, o ajuste é necessário quando se realizam estudos de estabilidade de um SEE, que requer a modelização de todos os seus componentes e ocorrem eventuais incompatibilidades entre os modelos dos sistemas de excitação fornecidos e os modelos disponíveis pela aplicação de simulação dinâmica que vai fazer os cálculos. Neste caso, o ajuste vai fornecer um modelo que apesar de não ser o modelo exato do sistema de excitação em questão, permite que a ferramenta de simulação dinâmica consiga replicar de forma fiável o seu comportamento dinâmico.

Em [3] são descritas algumas abordagens metodológicas para resolver o problema. Nessa dissertação, de uma forma simplificada, o algoritmo foi criado combinando uma análise visual da sensibilidade de cada parâmetro da função, quando modificado individualmente, com o método do gradiente que pretende minimizar o desvio quadrático médio entre as duas respostas. Na referida dissertação são também mencionadas técnicas, utilizadas na indústria, baseadas em ajuste manual do ganho, dos polos e dos zeros da função de transferência. Ambos os métodos, que vão ser referidos de forma mais extensa na secção 2.2, são orientados a modelos específicos e apenas fazem ajustes desses mesmos modelos.

No presente trabalho foi desenvolvida uma metodologia que permite a sua aplicação a qualquer tipo de modelos. Efetivamente, com o aumento da diversidade dos modelos e a sua constante evolução, torna-se interessante a criação de uma aplicação automática capaz de efetuar o ajuste para qualquer tipo de modelo. Esta metodologia parte do pressuposto que a função fornecida e a função a ajustar são matematicamente compatíveis, situação esta que nem sempre pode ser verdade.

1.2 - Objectivos

O trabalho aqui desenvolvido tem como objetivo a criação de uma aplicação capaz de fazer o ajuste do valor dos parâmetros de qualquer modelo matemático, minimizando a diferença da sua resposta no domínio das frequências, com a resposta de outro qualquer modelo matemático, utilizando uma meta-heurística do tipo EPSO. Tem também como

objetivo o teste da mesma aplicação para vários tipos de modelos com posterior verificação da qualidade dos resultados, recorrendo à comparação da resposta temporal dos dois modelos, o ajustado e o fornecido, utilizando a plataforma de simulação SIMULINK, disponível no MATLAB.

1.3 - Descrição geral dos restantes capítulos da tese

Os restantes capítulos desta tese foram organizados de acordo com a seguinte ordem de ideias:

- O capítulo 2 apresenta ao leitor sistemas de excitação de um ponto de vista teórico, assim como descreve outros trabalhos já desenvolvidos relacionados com o assunto em análise.
- O capítulo 3 explica os conceitos teóricos inerentes ao algoritmo de otimização utilizado para resolver o problema.
- O capítulo 4 apresenta matematicamente o problema de otimização desenvolvido e explica como este foi implementado.
- O capítulo 5 apresenta e discute os casos de estudo que foram testados.
- O capítulo 6 é dedicado às conclusões sobre o trabalho produzido.

Capítulo 2

Ajuste dos parâmetros para modelos de sistemas de excitação

2.1 - Sistemas de excitação

Os geradores síncronos são fundamentais para a existência de um SEE de grande dimensão, nomeadamente porque estes têm a possibilidade de controlar o seu fator de potência, consumindo ou produzindo energia reativa conforme desejado. Este controlo é feito pelos sistemas de excitação, que injetam uma corrente contínua no enrolamento do rotor da máquina síncrona, controlando desta forma o fluxo magnético no rotor, que por sua vez determina o fator de potência do gerador, permitindo controlar a razão entre a energia ativa e reativa sobre a potência total produzida pela máquina, que é uma variável essencial para o controlo e estabilidade de um SEE (toda a energia ativa e reativa produzida tem de ser consumida a cada instante de tempo para manter níveis de tensão e frequência estáveis). É apresentado na Figura 2.1 a arquitetura de um sistema de excitação.

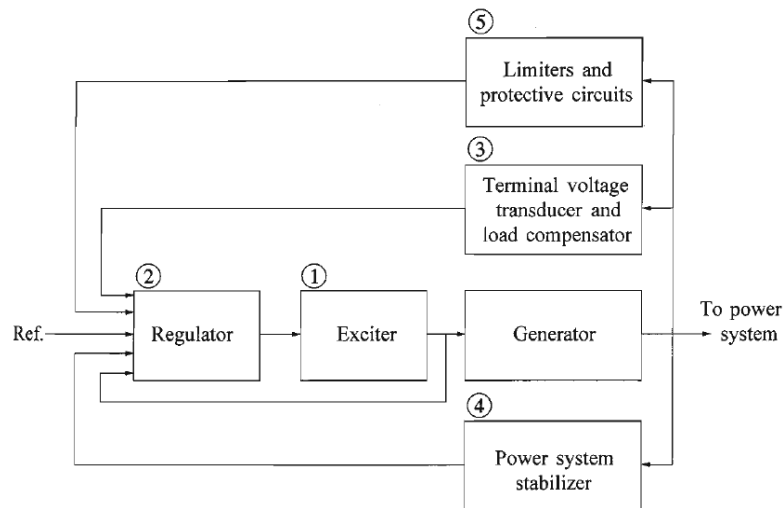


Figura 2.1 - Modelo do diagrama de blocos utilizado para este estudo, adaptado de [2]

Em termos funcionais, os sistemas de excitação são compostos por um regulador de tensão que estabelece uma malha de realimentação entre as grandezas elétricas (tensão e corrente) de saída do gerador síncrono e a excitatriz que alimenta o circuito rotórico do gerador, com corrente contínua. O regulador de tensão tem como função [14]:

- Manter a tensão aos terminais da máquina dentro dos seus valores estipulados.
- Regular o fator de potência das máquinas síncronas, dividindo a geração de potência reativa por todas as máquinas a operar em paralelo.
- Assegurar o sincronismo entre a máquina e o SEE.
- Reagir em situações de curto-circuito para tentar manter a máquina em sincronismo com o sistema.
- Amortecer oscilações de baixa frequência.

Em sistemas mais primitivos, a função do regulador de tensão era desempenhada pelo operador através da observação dos valores de tensão e corrente à saída do gerador. O ajuste era efetuado utilizando um reóstato de campo da excitatriz até obter o resultado pretendido. Este tipo de solução tem um tempo de resposta muito lento, prejudicando a estabilidade do SEE, pelo que, atualmente, os reguladores de tensão são eletrônicos e realizam controle automático de forma a aumentar a velocidade de resposta da malha de controle.

Em termos de construção, existem dois tipos de sistemas de excitação - os rotativos e os estáticos. Dentro dos sistemas de excitação rotativos existem dois tipos de sistema de excitação - os DC e os AC.

Nos sistemas de excitação DC, a excitatriz consiste geralmente num gerador de corrente contínua, montado no veio da máquina principal. A alimentação do rotor do gerador síncrono é geralmente efetuada a partir de sistema de escovas e anéis. Devido ao uso de escovas e anéis, este tipo de sistema de excitação necessitam de frequente manutenção e têm alguns

inconvenientes quando utilizados para alimentar geradores de elevada potência, nomeadamente:

- Elevadas correntes a baixa tensão, provocando desgaste nas escovas.
- Variações bruscas de carga que podem originar faíscas aos terminais do rotor.
- Dificuldade no acoplamento de geradores de corrente contínua a veios de geradores síncronos.

Nos sistemas de excitação AC, a excitatriz é constituída por um alternador montado no veio da máquina principal ligado a um circuito retificador, geralmente uma ponte de diodos, que por sua vez alimentava o rotor do gerador em corrente contínua.

Os sistemas de excitação mais recentes têm reposta mais rápida, consistindo em sistemas estáticos que utilizam eletrónica de potência para alimentar o rotor do gerador.

2.1.1 - Sistema de excitação DC1A

O modelo DC1A é um dos modelos standard, criados pelo IEEE para emular sistemas de excitação DC, suscetíveis para utilização em estudos de estabilidade de sistemas elétricos de energia de elevada dimensão.

O modelo tem como objetivo aproximar-se matematicamente à resposta temporal de um sistema de excitação DC real, quando ocorre qualquer perturbação no sistema elétrico a que se encontra ligado. Como se trata de uma aproximação, está estipulado pelo IEEE que o modelo apenas é válido para desvios de frequência que não ultrapassem $\pm 5\%$ de frequência nominal para oscilações do comportamento dinâmico dos geradores do sistema que não ultrapassem os 3 Hz. Devido ao imenso uso deste modelo pela indústria, ele é muitas vezes utilizado para representar outros tipos de sistemas de excitação, quando não existe muita informação disponível ou é necessário um modelo simplificado [1].

O modelo DC1A é descrito pelo diagrama de blocos da Figura 2.2.

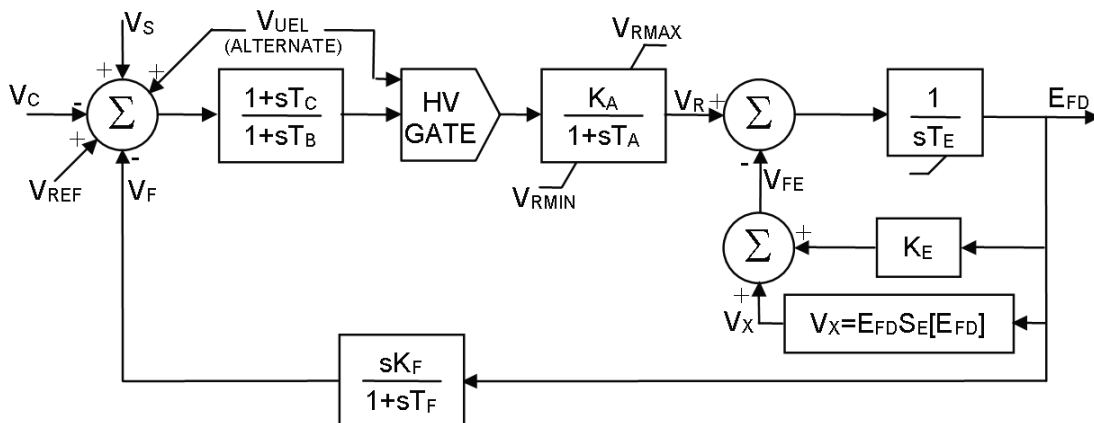


Figura 2.2 - Diagrama de blocos do modelo DC1A, adaptado de [1]

Sendo o modelo DC1A um modelo simplificado[1], não está no seu objetivo uma representação dos módulos físicos do sistema, mas sim emular a resposta temporal do sistema real quando sujeito a perturbações, dentro dos limites estipulados. Ainda assim, alguns dos blocos do modelo podem ser associados a componentes do sistema de excitação, descritos de seguida.

Um dos componentes que possui um bloco associado é o regulador de tensão. Do ponto de vista de modelização, o regulador de tensão é caracterizado por um ganho linear, por uma constante de tempo e limites de tensão, devido a saturação ou potência como pode ser observado na Figura 2.3.

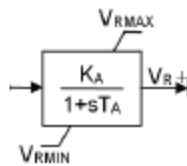


Figura 2.3 - Bloco simplificado que emula o regulador de tensão [1]

O sistema de excitação possui também funções de estabilização do sistema, modelizadas pelos blocos presentes na figura 2.4.

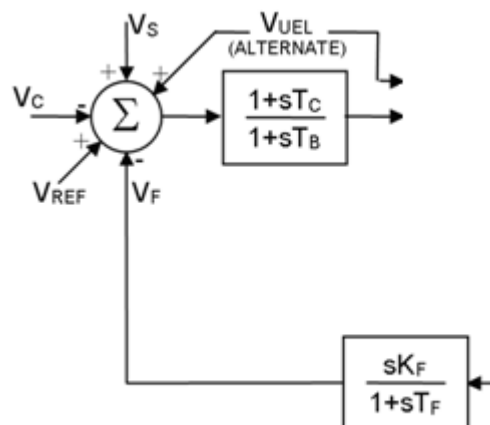


Figura 2.4 - Blocos que modelizam as funções de estabilização do regulador de tensão [1]

A excitatriz, por sua vez, do ponto de vista de modelização, necessita de um bloco adicional para além do ganho linear e da constante de tempo, uma vez que, tratando-se de um gerador de corrente contínua, está sujeita a saturação do fluxo magnético. A excitatriz é representada pelo conjunto de blocos da Figura 2.5.

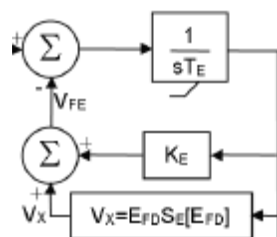


Figura 2.5 - Bloco simplificado que emula a excitatriz [1]

“ V_X ” modeliza a saturação da excitatriz, e “ S_E ” é uma função exponencial que modeliza a saturação do fluxo magnético da excitatriz. Como os casos de estudo desta dissertação envolvem simulação com o gerador síncrono em vazio, não ocorre saturação e “ V_X ” pode ser retirado do modelo. No entanto é de salientar que, uma vez que o algoritmo utilizado é uma meta-heurística do tipo EPSO, era possível utilizar um parâmetro de saturação não linear. Não foi utilizado por ser irrelevante para os resultados que se pretendiam produzir.

O modelo DC1A tem como parâmetros para ajuste K_A , K_E , K_F , T_A , T_B , T_C , T_E e T_F .

2.2 - Acerca de ajuste de parâmetros para modelos de sistemas de excitação

Neste capítulo referem-se outros trabalhos ou estudos relacionados com o tópico em análise no presente trabalho.

2.2.1 - Ajuste de parâmetros para modelos típicos de sistemas de excitação, recorrendo à resposta em frequência do modelo [3]

Em [3], são descritos os trabalhos de uma dissertação do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores onde foi desenvolvido um método para ajuste dos parâmetros do modelo DC1A por forma a obter uma resposta no domínio das frequências semelhante a outro modelo. O método é extensivamente descrito em [3] e segue essencialmente as seguintes ideias:

- Fazer uma inspeção visual observando como o DC1A se altera quando se alteram valores de combinações de parâmetros, por forma a isolar “formas de deslocamento” da resposta em frequência que se vão traduzir em expressões matemáticas em função das variáveis utilizadas.
- Aplicar o método do gradiente individualmente a essas “formas de deslocamento”, utilizando o erro quadrático médio para avaliar a qualidade da solução.

- A avaliação é realizada apenas sobre o módulo ou a fase da resposta, sendo escolhida a que produz melhores resultados.
- Após realizadas estas operações, realizar uma operação denominada “ajuste fino” que consiste no re-ajuste de alguns parâmetros, utilizando, como forma de avaliação, a diferença da resposta dos dois modelos para valores em torno de 0 Hz, possivelmente devido à obtenção de resultados que se afastavam muito para frequências em torno de 0 Hz.
- A solução final apresentada ao utilizador é a que apresenta o menor erro quadrático médio relativo.

De uma forma geral, o método vai tentando adaptar a resposta do DC1A ao modelo fornecido e à medida que vai encontrando problemas, o programador arranja uma forma de os eliminar, até chegar a uma resposta aceitável.

Embora o método obtenha resultados aceitáveis, a metodologia foi especificamente desenvolvida para os modelos dos casos de estudo apresentados. Uma vez que o IEEE apresenta 19 sistemas de excitação normalizados apropriados para estudos de estabilidade [1], a aplicação não tem versatilidade suficiente para cobrir todos os casos que podem ocorrer em estudos de estabilidade.

2.2.2 - Técnicas de melhoria de sistemas de excitação

Na secção 3.3 da dissertação atrás referida [3] são discutidas técnicas para melhoria de desempenho de um sistema de excitação. Embora elas se relacionem com o tópico de sistemas de excitação e possuam algumas semelhanças com a aplicação desenvolvida nesse trabalho, o método utilizado nada tem em comum com o método utilizado nesta dissertação. No entanto, caso a abordagem para com o problema não fosse através de uma meta-heurística, a otimização seria baseada nestas técnicas.

2.2.3 - Abordagem utilizada nesta dissertação

A abordagem utilizada nesta dissertação foi a criação de um algoritmo baseado em EPSO capaz de resolver o problema para o modelo DC1A, mas mantendo a versatilidade para outros possíveis modelos e tentando fazer a aplicação o mais genérico e “user friendly” possível. Foi escolhido o python, como linguagem de programação, devido a ser uma linguagem de programação “Open source” e devido a uma maior familiaridade do autor do presente trabalho com esta linguagem de programação. Inicialmente aplicou-se um algoritmo PSO ao problema, por ser mais simples, tornando-o depois num EPSO para lhe conferir propriedades auto-adaptativas e não ser necessária modificação manual dos parâmetros internos do algoritmo.

A qualidade dos resultados obtidos pelo algoritmo foi posteriormente testada utilizando o SIMULINK para obter a resposta dinâmica dos modelos fornecido e ajustado na sequência de uma perturbação pré-definida.

Capítulo 3

EPSO

3.1 - Introdução

O EPSO é um método híbrido de duas técnicas de otimização pertencentes à família das meta-heurísticas: EC (*Evolutionary Computation*) e PSO. Foi originalmente concebido pelo prof. Vladimiro Miranda [13] e possui inúmeras aplicações tanto em sistemas elétricos de energia como em outras áreas. O método confere a um algoritmo do tipo PSO características Auto adaptativas, pelo que não necessita de uma afinação manual dos seus parâmetros.

3.2 - Breve descrição de meta-heurísticas do tipo EC

Desde 1950 que são utilizados os princípios darwinianos de evolução para a resolução de problemas. Por volta de 1960 foram estabelecidos 3 métodos sobre este tópico que se desenvolveram separadamente por 15 anos, até à década de 90, onde foram unificadas sob o nome de “Evolutionary Computation” (EC).

Lawrence J. Fogel (1964)[4] apresentou a “evolutionary programming” (EP). Nos Estados Unidos, Holland [5] apresentou o seu método como “genetic algorithm” (GA), enquanto que Rechenberg & Schwefel (1971) [7] apresentaram as “Evolution Strategies” (ES) na Alemanha. Em meta-heurísticas do tipo EC, uma população de soluções candidatas (normalmente chamadas de “fenótipo”) a um determinado problema de otimização é evoluída para melhores soluções. Cada fenótipo possui um conjunto de propriedades que vão ser mudadas. Os operadores que geram a evolução são aplicados na forma de uma rotina, e cada iteração é normalmente designada de “geração”.

A população inicial é geralmente originada de forma aleatória, e a cada geração, a população é avaliada pelo operador “fitness”, que retorna o valor da função objetivo a ser otimizada. Os fenótipos vão ser então sujeitos a outro operador que faz uma seleção

estocástica que escolhe preferencialmente os mais adaptados e aplica o operador de mutação, que modifica as suas propriedades através de recombinação ou mutação aleatória para formar uma nova geração. A nova geração será utilizada na próxima iteração do algoritmo. A seleção não retira apenas os melhores de cada geração, com vista a manter diversidade.

Devido a estes operadores serem pesados do ponto de vista computacional, por vezes é apenas selecionada uma amostra aleatória da população.

O Critério de paragem é geralmente um numero máximo de iterações ou gerações, ou os fenótipos atingiram um valor da função objetivo aceitável (se for possível de averiguar um valor aceitável).

Na aplicação desenvolvida nesta dissertação a evolução é feita somando a cada propriedade “P” de cada fenótipo um número aleatório “A” que segue uma distribuição normal de média 0 e desvio padrão 1, multiplicada por uma taxa de mutação “Tm”, com valores tipicamente por volta de 0.001:

$$P = P + A * Tm \quad (3.1)$$

3.3 - Breve descrição de meta-heurísticas do tipo PSO

O algoritmo PSO foi originalmente introduzido por Kennedy e Eberheart [8] e visava simular o comportamento social de bandos de aves e cardumes de peixes em computador, sendo observado que o algoritmo efetuava otimização.

O PSO consiste numa população denominada de enxame, composta por partículas e que converge vai convergindo para o ótimo, atuando sobre a velocidade das partículas de uma forma orientada numa rotina. As partículas convergem para o ótimo utilizando ferramentas semelhantes às dos enxames de animais: Cooperação, memória e hábito (ou inércia), os componentes da regra do movimento.

Hábito (ou inércia) simplesmente impele a partícula a seguir a trajetória previamente tomada, enquanto a memória altera a velocidade da partícula no sentido do melhor valor encontrado por essa mesma partícula. Cooperação confere a todas as partículas do enxame velocidade no sentido do melhor valor encontrado por todo o enxame até ao momento, denominado de ótimo do enxame.

Sendo o PSO uma meta-heurística, não existe garantia de que o enxame convergiu para o ótimo global do espaço, existindo dois tipos possíveis de convergência:

- O ótimo do enxame convergiu para um valor aceitável, independentemente do movimento que as partículas têm quando isso ocorreu.
- Ocorreu um colapso do enxame em que todas as partículas convergiram para um ponto do espaço, sendo este o ótimo global ou não.

Para evitar o colapso do enxame foi utilizado outro fenómeno natural que se designa no presente trabalho espalhamento, que consiste na necessidade de as partículas se manterem

afastadas. Esta característica em conjunto com a característica hábito tem como objetivo evitar que o enxame fique "preso" num ótimo local, aumentando a eficiência da busca.

Sendo p um vetor de n partículas com m dimensões, v o vetor de velocidade dessas partículas, l o vetor de ótimos individuais das partículas, g o ótimo do enxame, cada instância de R é um número aleatório entre 0 e 1 de distribuição uniforme e f a função a minimizar, apresenta-se um possível algoritmo do tipo PSO:

- ❖ Para cada partícula $i=1, \dots, n$ fazer:
 - Inicializar a sua posição utilizando números aleatórios de distribuição uniforme com limites máximos e mínimos coincidentes com os limites do espaço de busca, sempre que possível. Inicializar o vetor velocidade com o valor 0, ou com valores aleatórios.
 - Colocar o ótimo individual l de cada partícula na posição inicial da partícula:
 $p_i \rightarrow l_i$
 - Se $f(p_i) < f(g)$ atualizar o ótimo do enxame: $p_i \rightarrow g$
- ❖ Enquanto um número máximo de iterações não for atingido:
 - Para cada partícula $i=1, \dots, n$ fazer:
 - Para cada parâmetro da "partícula i " $j=1, \dots, m$ fazer:
 - $v_{ij} \leftarrow \varphi_0 v_{ij} + \varphi_1 R(l_{ij} - p_{ij}) + \varphi_2 R(g_j - p_{ij})$
 - Atualizar a posição da partícula $p_i \rightarrow p_i + v_i$
 - Se $f(p_i) < f(l_i)$ atualizar o ótimo individual: $p_i \rightarrow l_i$
 - Se $f(l_i) < f(g)$ atualizar o melhor global: $l_i \rightarrow g$

Os parâmetros φ_1 , φ_2 e φ_3 são denominados de pesos e definidos pelo utilizador. Eles controlam a eficácia do PSO. Ao aumentar o seu valor a busca dá saltos de posição maiores entre iterações, gerando uma busca menos minuciosa. Ao diminuir o seu valor, diminui-se os saltos de posição, tornando a busca mais minuciosa.

Sendo o espalhamento um extra definido pelo autor, ele não faz parte do algoritmo básico, pelo que será explicado mais à frente no capítulo 4.

3.4 - Breve descrição de meta-heurísticas do tipo EPSO

Se utilizarmos o algoritmo PSO atrás descrito e utilizarmos os princípios de EC para evoluir os parâmetros que têm de ser definidos pelo utilizador no PSO, obtemos o algoritmo EPSO:

- ❖ Para cada partícula $i=1, \dots, n$ fazer:
 - Inicializar a sua posição utilizando números aleatórios de distribuição uniforme com limites máximos e mínimos coincidentes com os limites do espaço de busca, sempre que possível. Inicializar o vetor velocidade com o valor 0, ou com valores aleatórios.
 - Criar um clone das partículas $p_i \rightarrow p_{ci}$ e da sua velocidade $v_i \rightarrow v_{ci}$.

- Adicionar os pesos iniciais a um vetor de pesos $[\varphi_0, \varphi_1, \varphi_2] \rightarrow \varphi_i$ e criar um clone $[\varphi_0, \varphi_1, \varphi_2] \rightarrow \varphi_{ci}$.
- Colocar o ótimo individual l de cada partícula na posição inicial da partícula:
 $p_i \rightarrow l_i$
- Se $f(p_i) < f(g)$ atualizar o ótimo do enxame: $p_i \rightarrow g$
- ❖ Enquanto um número máximo de iterações não for atingido:
 - Para cada partícula $i=1, \dots, n$ fazer:
 - Mutar os pesos clone: $\varphi_{i0} + \text{TaxMuta} \cdot \text{Gauss}(\sigma = 0, \mu = 1) \rightarrow \varphi_{ci0}$
 - Para cada dimensão $j=1, \dots, m$ fazer:
 - $v_{ij} \leftarrow \varphi_{i0}v_{ij} + \varphi_{i1}R(l_{ij} - p_{ij}) + \varphi_{i2}R(g_j - p_{ij})$
 - $v_{cij} \leftarrow \varphi_{ci0}v_{ij} + \varphi_{ci1}R(l_{ij} - p_{cij}) + \varphi_{ci2}R(g_j - p_{cij})$
 - Atualizar a posição da partícula $p_i \rightarrow p_i + v_i$ e do seu clone $p_{ci} \rightarrow p_i + v_{ci}$.
 - Se $f(p_{ci}) < f(p_i)$:
 - $p_{ci} \rightarrow p_i$
 - $\varphi_{ci} \rightarrow \varphi_i$
 - Se $f(p_i) < f(l_i)$: atualizar o ótimo individual: $p_i \rightarrow l_i$
 - Se $f(l_i) < f(g)$: atualizar o ótimo do enxame: $l_i \rightarrow g$

Em que $\text{Gauss}(\sigma = 0, \mu = 1)$ é um número aleatório com uma distribuição normal de valor esperado 0 e variância 1 e TaxMuta é a taxa de mutação, com valores tipicamente a rondar os 0,001. Os valores iniciais dos pesos são geralmente um número entre 0 e 1, tipicamente por volta de 0,5.

Hábito + Memória + Cooperação = Equação do movimento

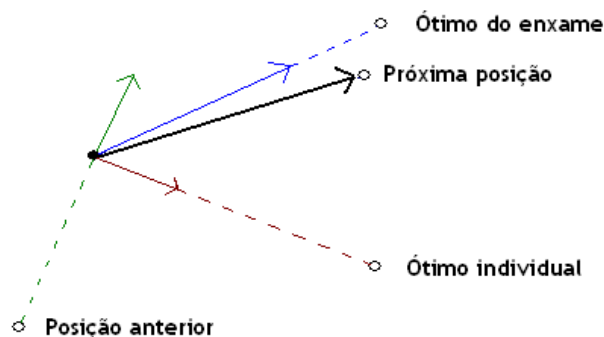


Figura 3.1 - Representação da equação do movimento e os seus componentes sobre uma partícula

Capítulo 4

Metodologia desenvolvida

4.1 - Formulação do problema de otimização

Este capítulo vai descrever o problema de otimização desta dissertação de uma forma detalhada. Nele é retratado o problema do ajuste dos parâmetros do modelo DC1A para a sua resposta se aproximar da resposta do modelo fornecido no caso base.

Para começar, é necessário obter a resposta dos diagramas de blocos sobre a forma de uma função, tanto do modelo DC1A como do modelo fornecido. Revendo a teoria de sistemas realimentados [10], sabemos que um sistema realimentado de malha fechada do tipo da Figura 4.1 tem a seguinte função de transferência:

$$\frac{Y}{X} = \frac{G(s)}{1+G(s) \times H(s)} \quad (4.1)$$

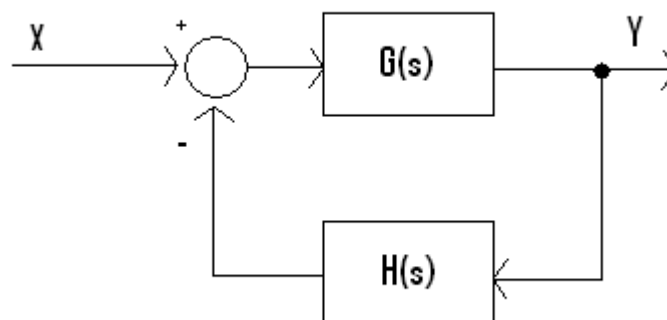


Figura 4.1 - Diagrama de blocos genérico de uma função de transferência com malha de realimentação

Devido à construção do modelo DC1A apenas apresentar uma aproximação do sistema real, o sistema de excitação apenas é corretamente representado no domínio de frequência entre 0 e 3 Hz [9] e por essa razão, o ajuste será feito com as respostas dos diagramas de blocos calculadas apenas nesse domínio. Os diagramas de blocos usam $s = j\omega = j \cdot 2\pi \cdot f$, pelo que o domínio de análise do operador s corresponde a $C \in [0; 2\pi \times 3] \text{ rad/s}$. Na figura 4.2 podem ser observados os diagramas de blocos utilizados no simulink para o modelo a ajustar DC1A e para o modelo fornecido como caso de estudo base.

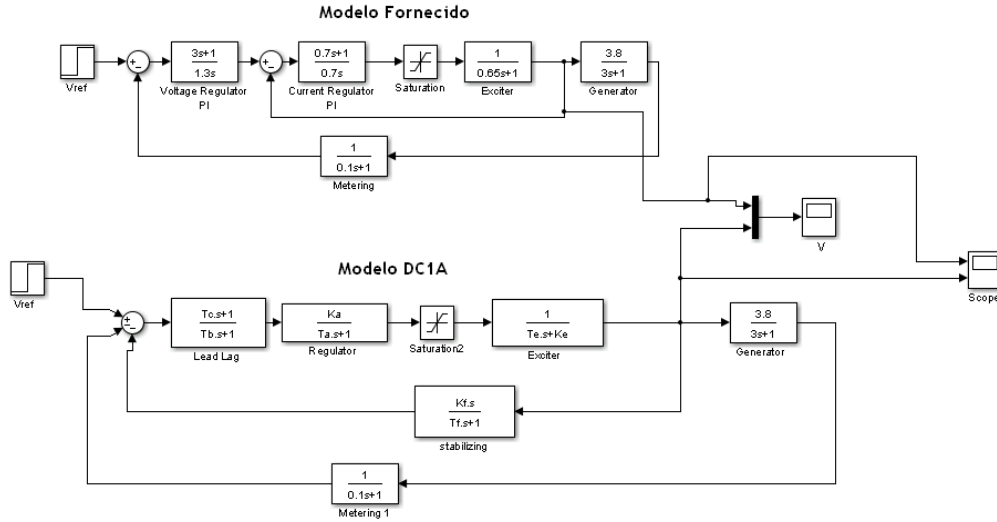


Figura 4.2 - Diagrama de blocos do modelo a ajustar DC1A e do modelo fornecido como caso base em simulink

Os cálculos serão feitos retirando todos os blocos que os dois diagramas possuem em comum (bloco do gerador e de medição). O problema tem 8 variáveis armazenadas num vetor x com um domínio D , que consistem nos parâmetros do modelo DC1A. O domínio D vai ser $[0; +\infty]$ para todos os parâmetros, com a exceção de T_B, T_E, T_F e T_A que vão ter um domínio de $[0,005; +\infty]$. Desta forma, o modelo a ajustar nesta dissertação é o DC1A, consistindo na seguinte função de transferência:

$$g(s, x) = \frac{\frac{s.T_C+1}{s.T_b+1} \times \frac{K_a}{s.T_a+1} \times \frac{1}{s.T_e+K_e}}{1 + \frac{s.K_f}{s.T_f+1} \times \frac{s.T_C+1}{s.T_b+1} \times \frac{K_a}{s.T_a+1} \times \frac{1}{s.T_e+K_e}} \quad (4.2)$$

Que, neste caso de estudo base, terá de se ajustar à seguinte função de transferência fornecida:

$$h(s) = \frac{3s+1}{1,3s} \times \frac{\frac{0,7s+1}{0,7s} \times \frac{1}{0,65s+1}}{1 + \frac{0,7s+1}{0,7s} \times \frac{1}{0,65s+1}} \quad (4.3)$$

Define-se então o problema do ajuste do modelo DC1A (designado de $g(s, x)$) a um outro modelo fornecido (designado $h(s)$):

$$\max f = - \int_C \|g(s, x) - h(s)\| \partial s, x \in D, C \in [0; 2\pi 3] \quad (4.4)$$

É de notar que f corresponde ao negativo da área entre as duas funções, pelo que $f \leq 0$ e será 0 quando as respostas das duas funções forem coincidentes para o domínio C . Na figura 4.3 é possível visualizar gráficamente um exemplo de modelo fornecido e ajuste DC1A, e a respetiva função objetivo calculada para os mesmos.

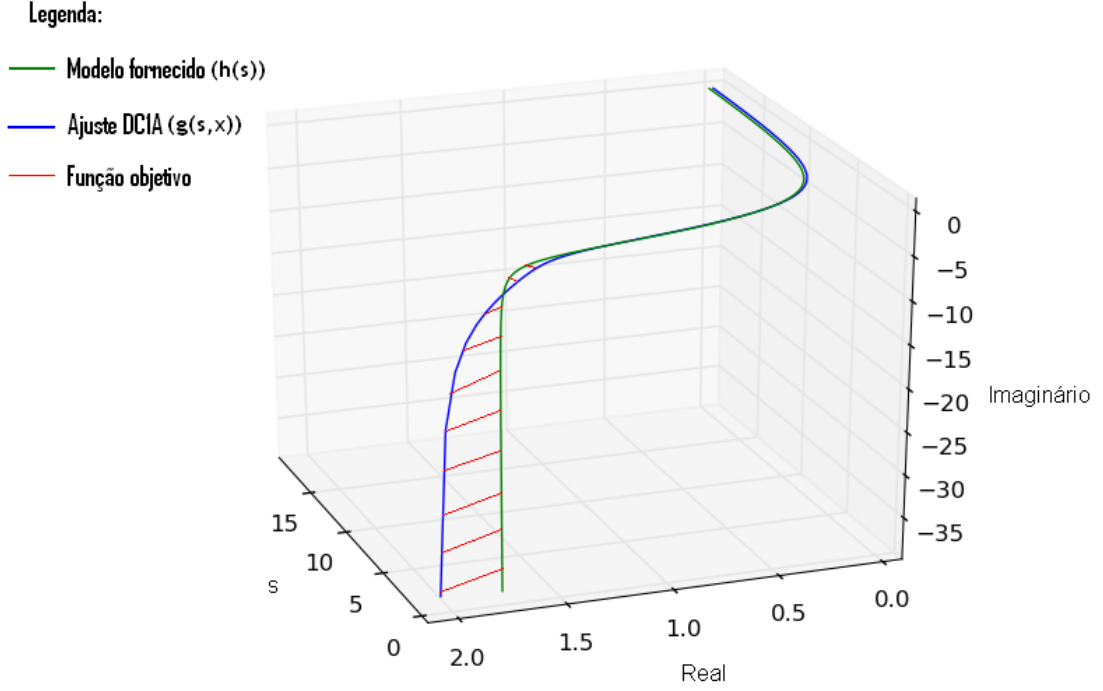


Figura 4.3 - Representação do ajuste da resposta $g(s,x)$ a $h(s)$ com s no domínio C utilizando a aplicação desenvolvida nesta dissertação, com a função objetivo salientada

Como em computação é impossível fazer cálculos com funções contínuas, é necessário retirar um sub-conjunto discreto com n componentes de C , por exemplo definido da seguinte forma:

$$C' = \left[0 + \frac{1 \times 3 \times 2\pi}{n}; 0 + \frac{2 \times 3 \times 2\pi}{n}; \dots; 0 + \frac{n \times 3 \times 2\pi}{n} \right] \quad (4.5)$$

Originando a nova função objetivo baseada numa soma discreta:

$$\max f' = - \sum_{s \in C'} (\|g(x,s) - h(s)\|) \quad , x \in D \quad (4.6)$$

Este problema é uma aproximação do problema contínuo e apenas garante que as funções coincidam nos pontos pertencentes a C' , pelo que se n for maior, a representação é mais exata mas também mais pesada computacionalmente.

Devido ao sucesso dos resultados produzidos com a função objetivo presente em (4.6), não foi necessário refletir sobre funções objetivo alternativas.

4.2 - Implementação do EPSO

4.2.1 - Geral

Nesta secção, explica-se como se adaptou o algoritmo EPSO ao problema de otimização descrito na secção 4.1. O algoritmo implementado neste problema é semelhante ao exposto na secção 3.4, com as seguintes adaptações:

- Cada partícula vai ter como variáveis os parâmetros do modelo a ajustar. A sua inicialização será feita com números aleatórios de distribuição uniforme gerados entre limites típicos para as variáveis em questão sempre que possível. No caso do DC1A esses limites foram retirados de [12].
- A avaliação vai ser feita com base na equação (4.6), pelo que vai ser necessário calcular a resposta em frequência provocada por cada partícula no modelo DC1A.
- Quando é realizada a avaliação, as soluções são penalizadas caso contenham alguma variável fora dos limites permitidos, definidos segundo o domínio D, definido na secção 4.1. Serão penalizadas proporcionalmente ao número de restrições quebradas.

4.2.2 - Colapso do enxame

Devido ao problema de colapso de enxame mencionado na secção 3.3 ocorrer frequentemente para este tipo de problema, foi necessário pensar numa estratégia para impedir o enxame de convergir para ótimos locais. Uma técnica atualmente utilizada para evitar o colapso do enxame é a implementação de colisões elásticas entre partículas, isto é, quando duas partículas estão no mesmo ponto ocorre um embate que vai alterar o sentido da sua velocidade, dependendo do ângulo de embate, mantendo o momento linear, isto é, mantendo o mesmo módulo da velocidade. Esta técnica, apesar de ser interessante em conceito, não produziu resultados muito bons no problema de otimização explorado nesta dissertação.

Outra técnica consiste em gerar vários enxames, em paralelo, em locais diferentes do espaço, com uma pequena probabilidade de comunicação uns com os outros (transmitir o seu ótimo para o outro). Esta técnica baseia-se no pressuposto de que, caso um enxame colapse, o outro continua a procurar um ótimo aceitável e caso os dois colapsem, eventualmente vai ocorrer comunicação e um dos enxames colapsados vai começar a deslocar-se no sentido do outro. É fácil de perceber que é uma técnica pesada computacionalmente e desperdiça recursos computacionais pois não impede, de facto, o colapso de um enxame, apenas espera que os outros continuem a procurar, eventualmente “desencalhando” o enxame colapsado

com comunicação. Mais uma vez, como no problema desta dissertação existem muitos pontos possíveis para o colapso do(s) enxame(s) devido aos modelos serem funções polinomiais de elevado grau com números complexos, esta técnica não é à partida muito interessante.

Desta forma, nesta dissertação foi necessário pensar numa nova forma de fazer o enxame não apenas evitar os ditos “pontos de colapso” como afastar-se deles, após ter encontrado esse ótimo, local ou não.

O problema foi resolvido acrescentando aos pesos, um novo peso denominado de “espalhamento”. Esse peso também é mutado como os restantes, pelo que se vai adaptando às necessidades do enxame.

Um novo componente é então adicionado à equação de velocidade do algoritmo que impele cada partícula a afastar-se do ponto médio (ou centro de massa) do enxame. Esse afastamento no entanto só deve ser “de elevado módulo” quando a partícula se encontra muito próxima do centro de massa (o que ocorre quando as partículas estão todas juntas), devendo ser muito pequeno quando a partícula se encontra relativamente longe. Isso pode ser manipulado tornando o módulo desse componente da velocidade inversamente proporcional à distância da partícula ao ponto médio do enxame, resultando na nova equação de velocidade:

$$v_{ij} \leftarrow \varphi_{i0} v_{ij} + \frac{e_i}{D_i} R(p_{ij} - m_j) + \varphi_{i1} R(l_{ij} - p_{ij}) + \varphi_{i2} R(g_j - p_{ij}) \quad (4.7)$$

Em que e_i é o peso do espalhamento para a partícula i , D_i a distância da partícula i ao ponto médio do enxame e m_j o parâmetro j do ponto médio do enxame.

4.2.3 - Verificação da solução

Como a partir do valor da função objetivo é difícil verificar se a solução obtida é satisfatória ou não, a melhor solução demonstrada pela metodologia desenvolvida deverá ser avaliada visualmente através de uma série de gráficos que foram implementados na aplicação desenvolvida:

- Diagrama de Bode do módulo e da fase da resposta em frequência do último ótimo global encontrado, comparado com o diagrama de Bode do módulo e a fase da resposta do modelo fornecido (ver exemplos na figura 4.4 e 4.5);
- Gráfico tridimensional contendo a parte real e imaginária das respostas dos modelos para cada valor de s (ver exemplo na figura 4.6);
- Gráfico contendo a soma das distâncias das partículas ao ponto médio do enxame (para ter uma noção da dispersão do enxame) ao longo das iterações (ver exemplo na figura 4.7).
- Evolução do valor da função objetivo do ótimo global ao longo das iterações (ver exemplo na figura 4.8);

Cabe ao utilizador determinar se o ótimo obtido é suficiente. É de notar que o método pode nunca encontrar uma adaptação suficientemente boa, devido à não existência de uma boa adaptação para os modelos a aproximar inseridos.

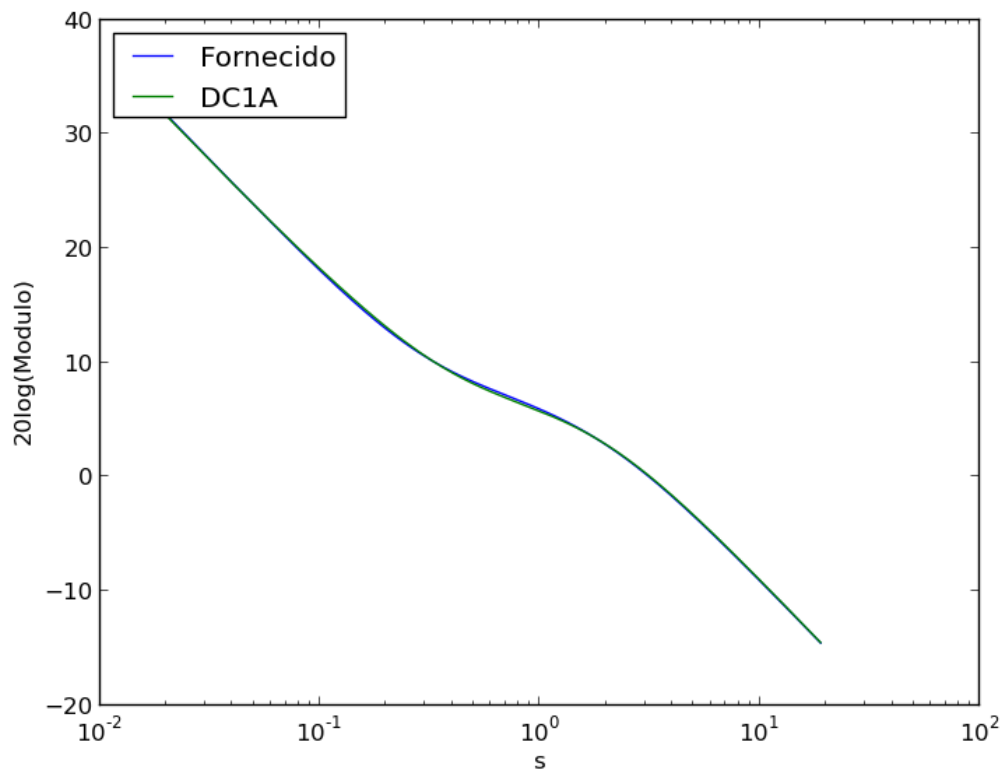


Figura 4.4 - Diagrama de Bode do módulo da resposta do ótimo global obtido (DC1A) e do modelo fornecido

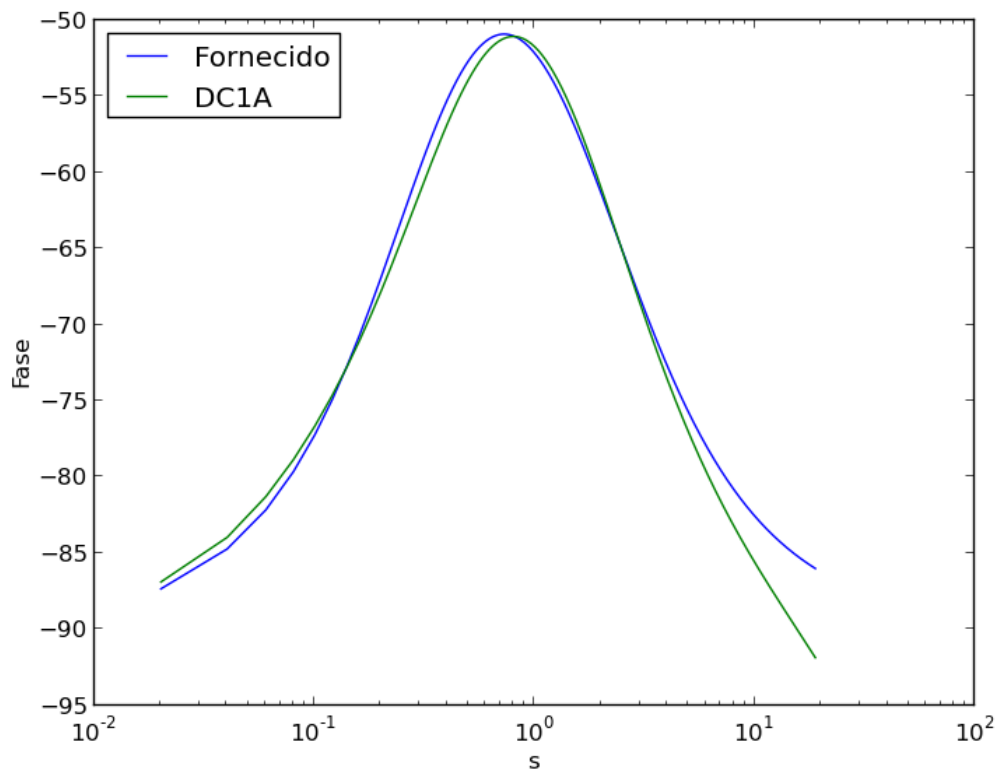


Figura 4.5 - Diagrama de Bode da fase da resposta do ótimo global obtido (DC1A) e do modelo fornecido

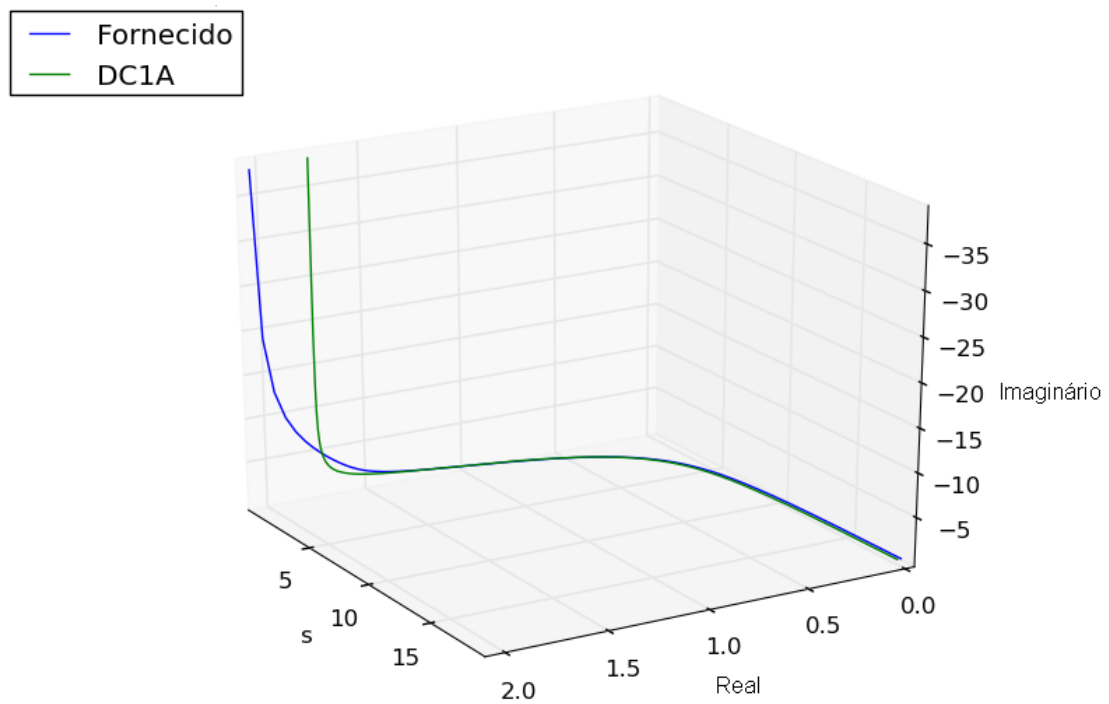


Figura 4.6 - Gráfico tridimensional da resposta do ótimo global obtido (DC1A) e do modelo fornecido

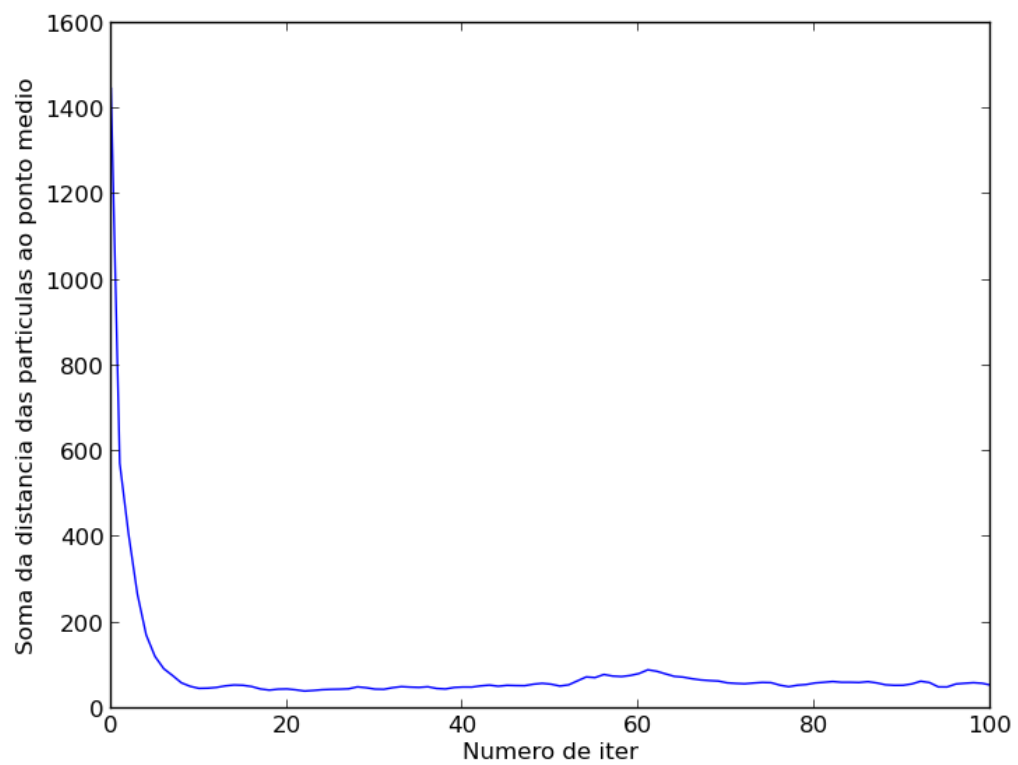


Figura 4.7 - Soma das distâncias das partículas ao ponto médio do enxame em cada iteração

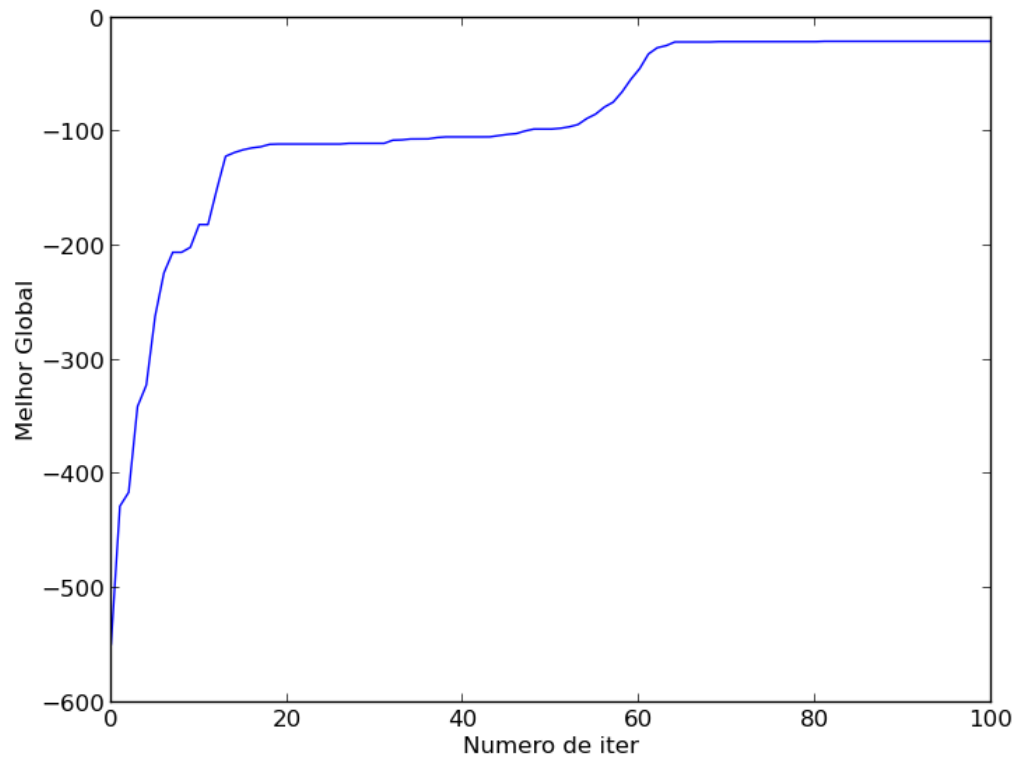


Figura 4.8 - Evolução do valor da função objetivo do ótimo global ao longo das iterações

Capítulo 5

Casos de estudo

5.1 - Modelos fornecidos

Para testar a aplicação desenvolvida, foram fornecidos dois diagramas de blocos alternativos para o modelo fornecido e vários casos de estudo para cada um dos modelos, através da consideração de valores alternativos para os parâmetros do modelo. Os modelos vão ser apresentados conforme foram utilizados no simulink. A inicialização do enxame ocorreu entre 0 e 10 para todos os parâmetros, e foram atribuídos limites mínimos de 0,005 aos parâmetros para não inviabilizar a simulação da solução em simulink, pois ganhos ou constantes de tempo negativas podem criar problemas de instabilidade no modelo dinâmico.

5.1.1 - Modelo fornecido 1

O primeiro modelo fornecido para os casos de estudo corresponde ao que se apresenta na parte superior da seguinte figura:

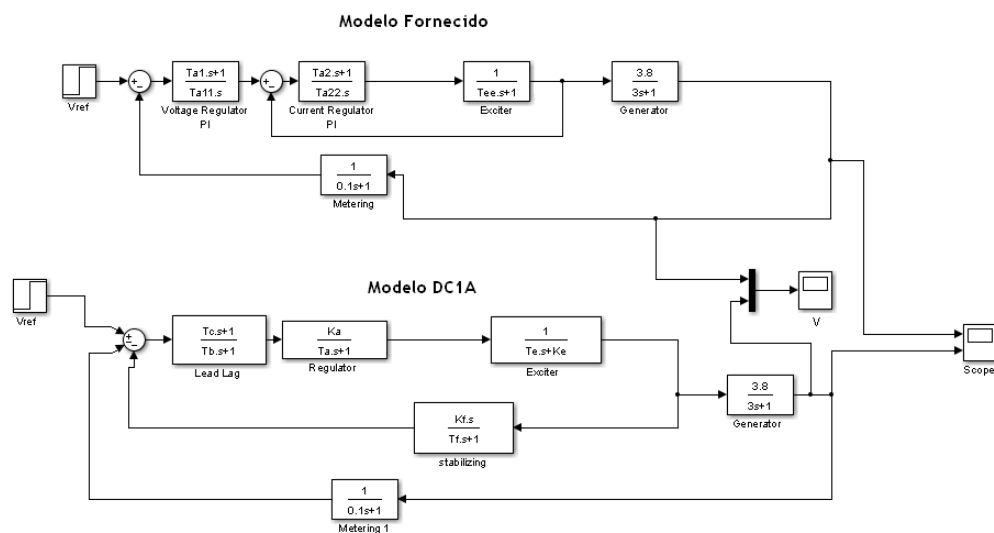


Figura 5.1 - Esquema de simulink utilizado para simular o primeiro modelo dos casos de estudo

O diagrama de blocos no topo corresponde ao modelo fornecido acoplado a um modelo de um gerador em vazio, e um bloco de medição. Por baixo é possível ver o modelo do DC1A, também acoplado ao gerador em vazio e ao bloco de medição para efeitos de simulação. A perturbação considerada na simulação dinâmica, para comparar a resposta dos modelos

(fornecido e ajustado), consistiu num aumento em degrau, aos 1 segundos, na tensão de referência (sinal Vref) com uma amplitude de 0,01 p.u. A resposta temporal comparada consistiu no valor eficaz da tensão aos terminais do gerador (sinal V).

5.1.2 - Modelo 2

O Segundo modelo fornecido para os casos de estudo corresponde ao que se apresenta na parte superior da seguinte figura:

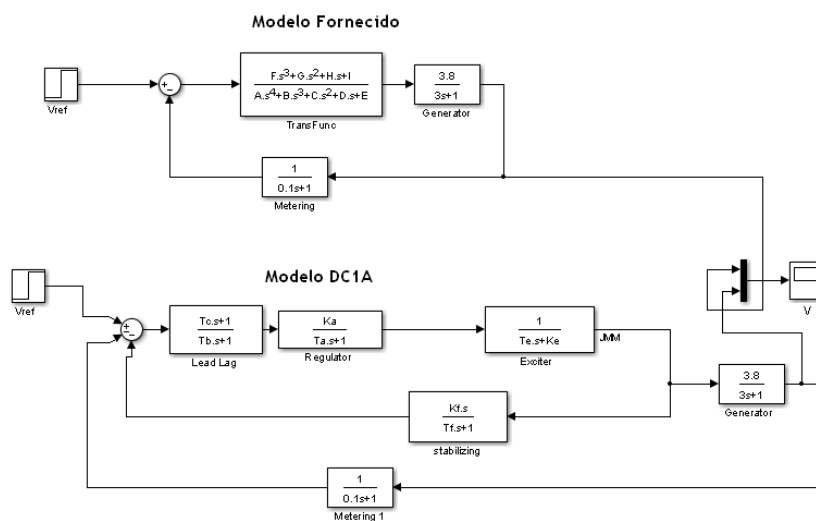


Figura 5.2 - Esquema simulink utilizado para validar o ajuste obtido para o segundo modelo fornecido

Mais uma vez, o diagrama do topo corresponde ao modelo fornecido, acoplado a um gerador em vazio e um bloco de medição, enquanto que em baixo temos o modelo DC1A a ajustar também acoplado a um gerador em vazio e bloco de medição. Nesta simulação dinâmica, manteve-se a mesma perturbação e visualização de respostas temporais já descritas para o modelo anterior.

5.2 - Casos de estudo do modelo 1

Para o modelo 1 foram fornecidos os seguintes casos de estudo.

Tabela 5.1 - Casos de estudo do modelo 1

Parâmetros do modelo 1	Caso Base	Caso 1	Caso 2	Caso 3
Ta1	3	1	3	8
Ta11	1,3	0,9	1,3	3
Ta2	0,7	0,7	1,3	1,3
Ta22	0,7	1,3	1,3	0,7
Tee	0,65	1	0,65	0,9

5.2.1 - Caso Base

Após a execução do programa desenvolvido para o modelo 1 com os valores do caso base que se apresentam na tabela 5.1 e uma inicialização dos parâmetros entre $[0,10]$, foi obtido o ajuste para os parâmetros do modelo DC1A presentes na tabela 5.2.

Tabela 5.2 - Valores dos parâmetros DC1A obtidos para o caso base

Ka	15,6313456862
Ke	8,96963887883e-06
Kf	1,41812634607
Ta	19,8409907472
Tb	0,00500002189813
Tc	8,94318919268
Te	2,02174612387
Tf	3,66160980181

Este ajuste originou os diagramas de bode, para o módulo e a fase das respostas dos dois modelos no domínio das frequências, apresentados nas figuras 5.3 e 5.4.

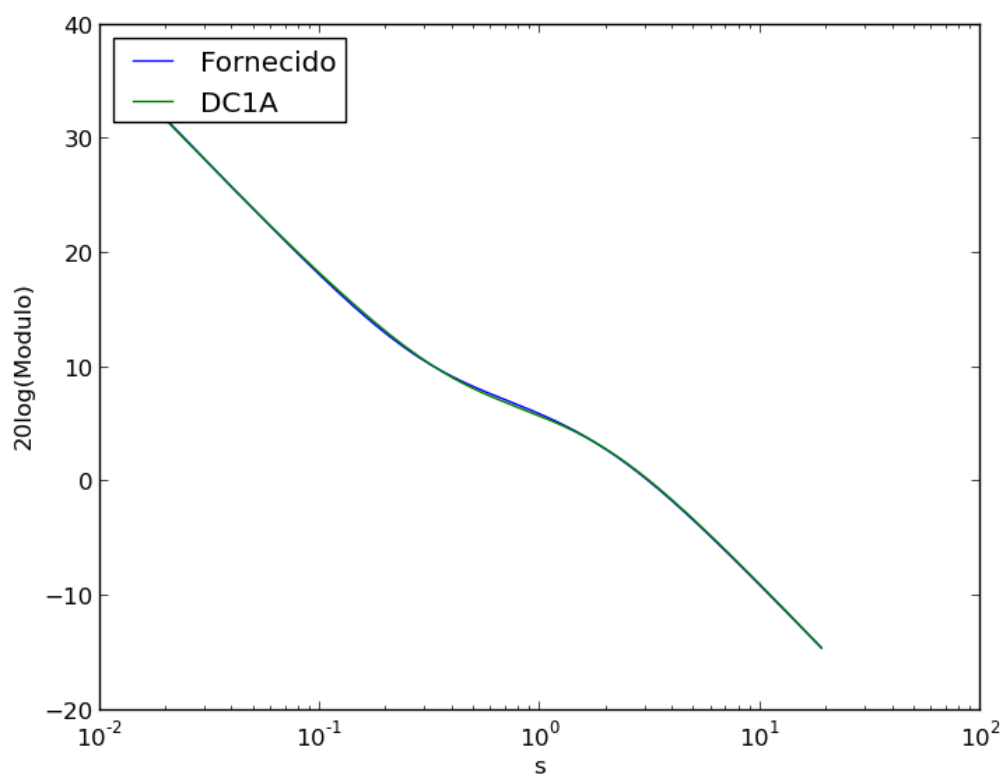


Figura 5.3 - Diagramas de bode - Módulo (solução obtida para o caso base)

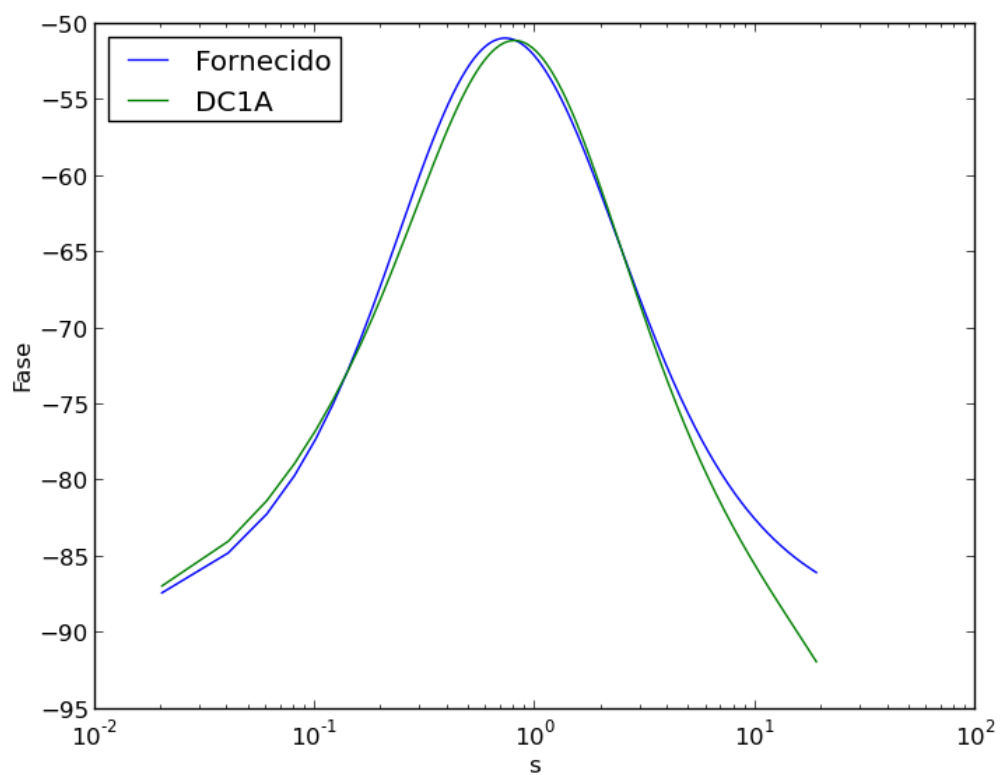


Figura 5.4 - Diagramas de bode - Fase (solução obtida para o caso base)

E a seguinte resposta temporal obtida para V, utilizando o modelo 1 no simulink (ver figura 5.1), presente na figura 5.5.

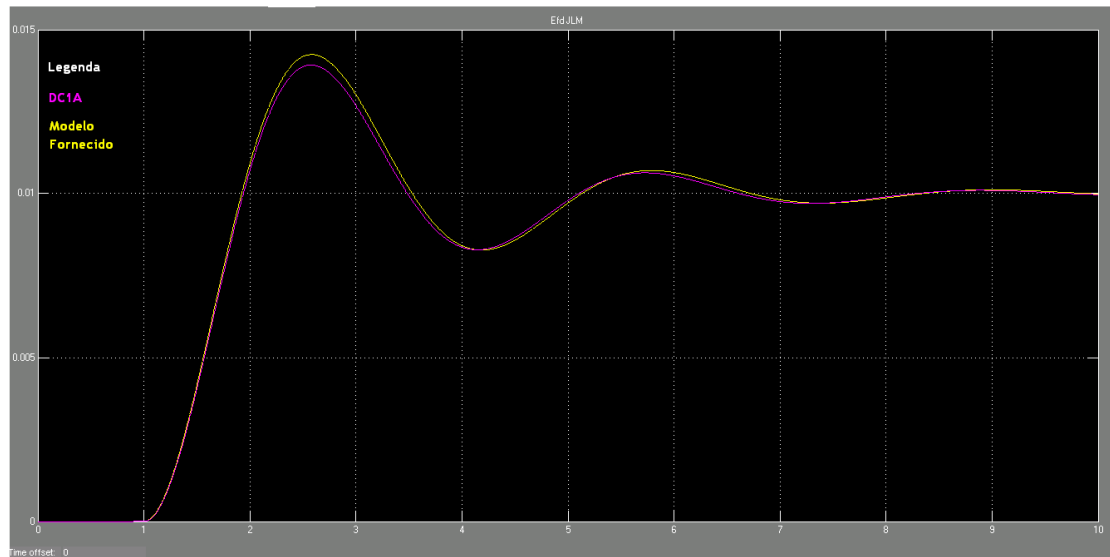


Figura 5.5 - Resposta temporal de V (solução obtida para o caso base)

Os resultados obtidos e apresentados da figura 5.3 à figura 5.5 confirmam a grande qualidade alcançada pela metodologia desenvolvida neste trabalho, para este caso de estudo.

5.2.2 - Caso de estudo 1

Após a execução do programa desenvolvido para o modelo 1 com os valores do caso de estudo 1 que se apresentam na tabela 5.1 e uma inicialização dos parâmetros entre $[0,10]$, foi obtido o ajuste para os parâmetros do modelo DC1A presentes na tabela 5.3.

Tabela 5.3 - Valores dos parâmetros para o modelo DC1A obtidos para o caso de estudo 1

Ka	10,392942525091389
Ke	0,0006502968444271584
Kf	0,8129367092739989
Ta	4,920588461825231
Tb	0,01446010810413216
Tc	0,38568786644526315
Te	0,8381468030616592
Tf	0,0325886408578548

Este ajuste originou os diagramas de bode, para o módulo e a fase das respostas dos dois modelos no domínio das frequências, apresentados nas figuras 5.6 e 5.7.

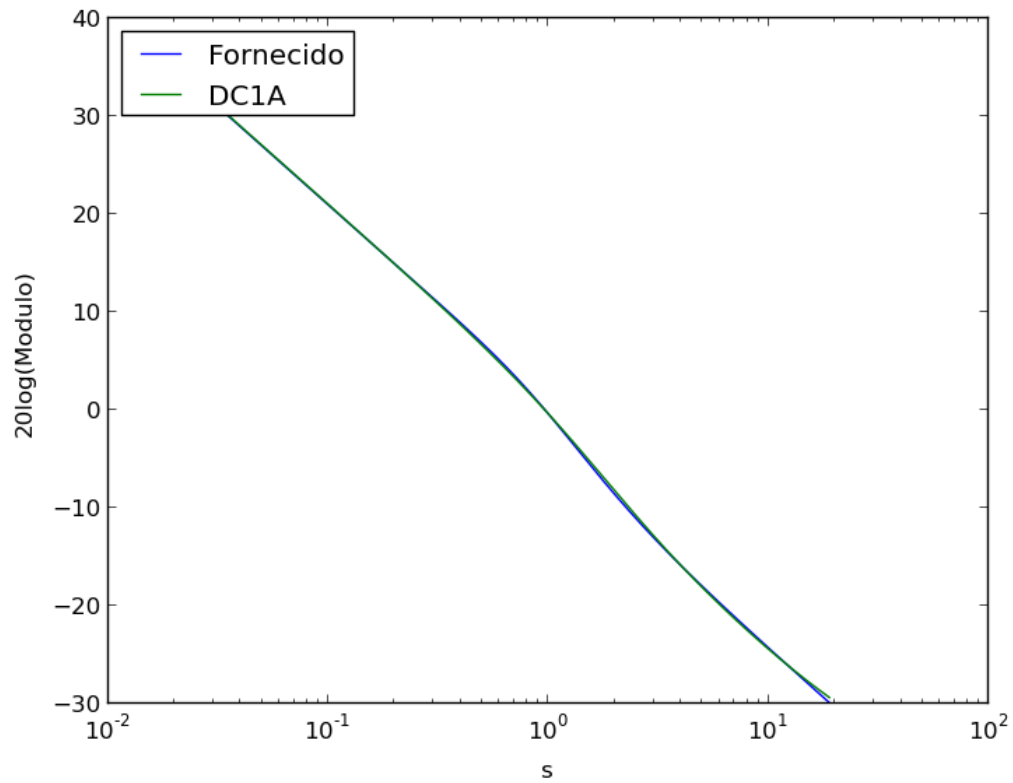


Figura 5.6 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 1)

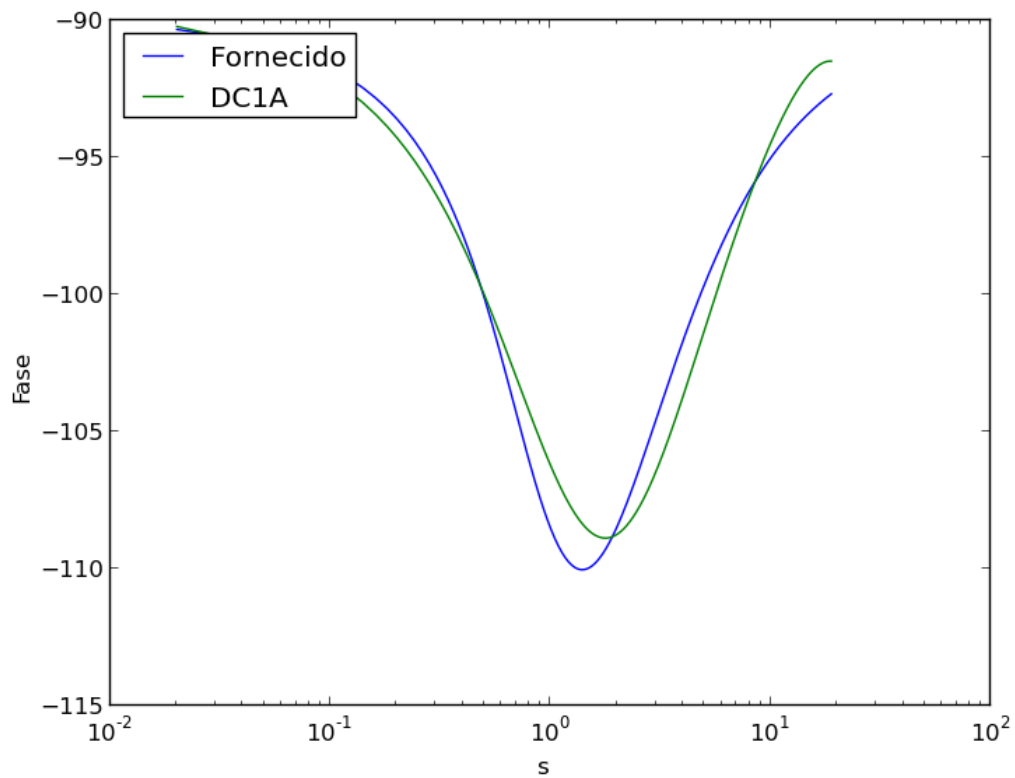


Figura 5.7 - Diagramas de bode - Fase (solução obtida para o caso de estudo 1)

E a seguinte resposta temporal obtida para V , utilizando o modelo 1 no simulink (ver figura 5.1), presente na figura 5.8.

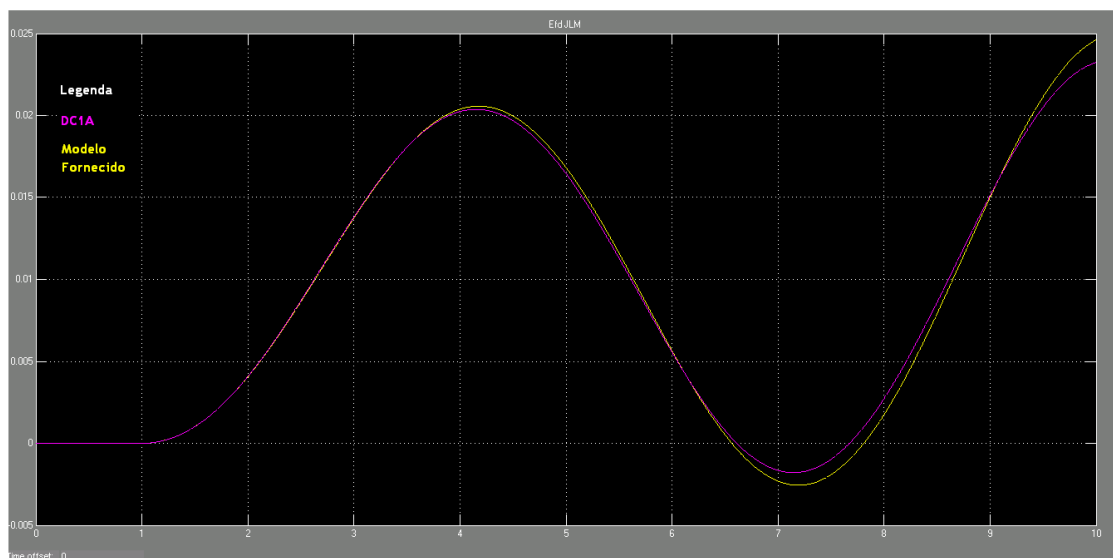


Figura 5.8 - Resposta temporal de V (solução obtida para o caso de estudo 1)

Como se pode observar nesta figura, mesmo para situações em que a resposta dinâmica do modelo fornecido é instável, a metodologia desenvolvida foi capaz de encontrar uma solução de ajuste com boa qualidade.

5.2.3 - Caso de estudo 2

Após a execução do script para o modelo 1 com os valores do caso de estudo 2 (ver tabela 5.1) e uma inicialização dos parâmetros entre $[0,10]$, foi obtido o seguinte ajuste para os parâmetros do modelo DC1A.

Tabela 5.4 - Valores dos parâmetros para o modelo DC1A obtidos para o caso de estudo 2

Ka	3,5744661681844123
Ke	0,0020406583672078087
Kf	0,5718345726790981
Ta	1,4592389619383774
Tb	0,005129117547776939
Tc	3,6092999575748355
Te	2,4513589566389116
Tf	1,2983779270380156

Este ajuste originou os diagramas de bode, para o módulo e a fase das respostas dos modelos no domínio das frequências, que se apresentam na figura 5.9 e figura 5.10.

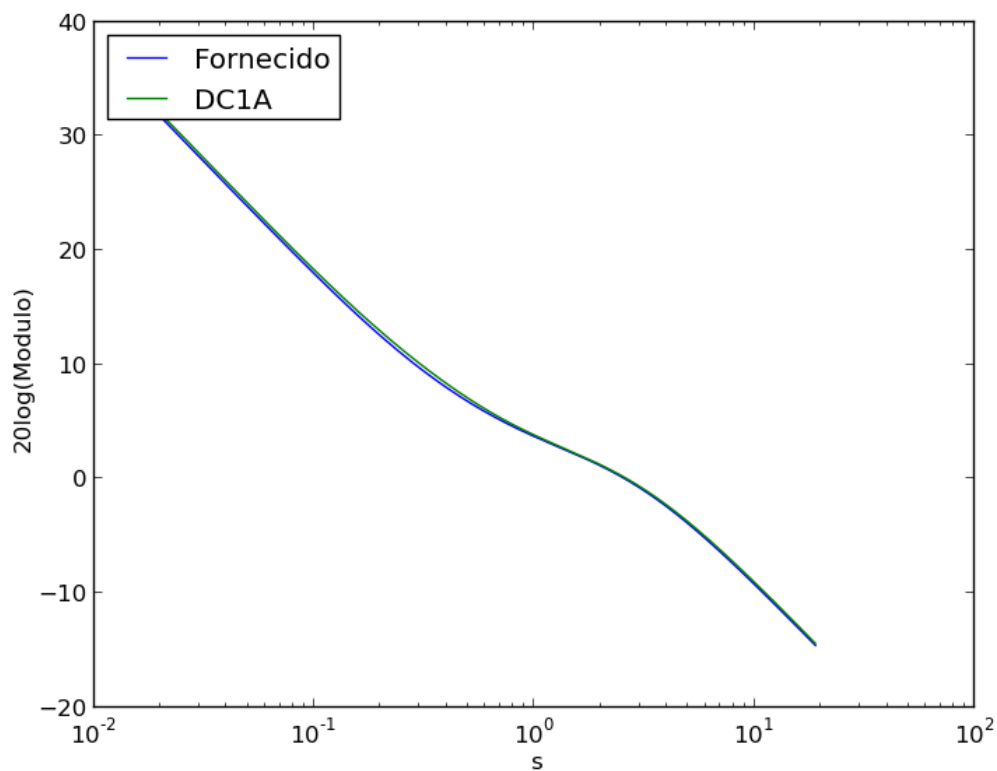


Figura 5.9 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 2)

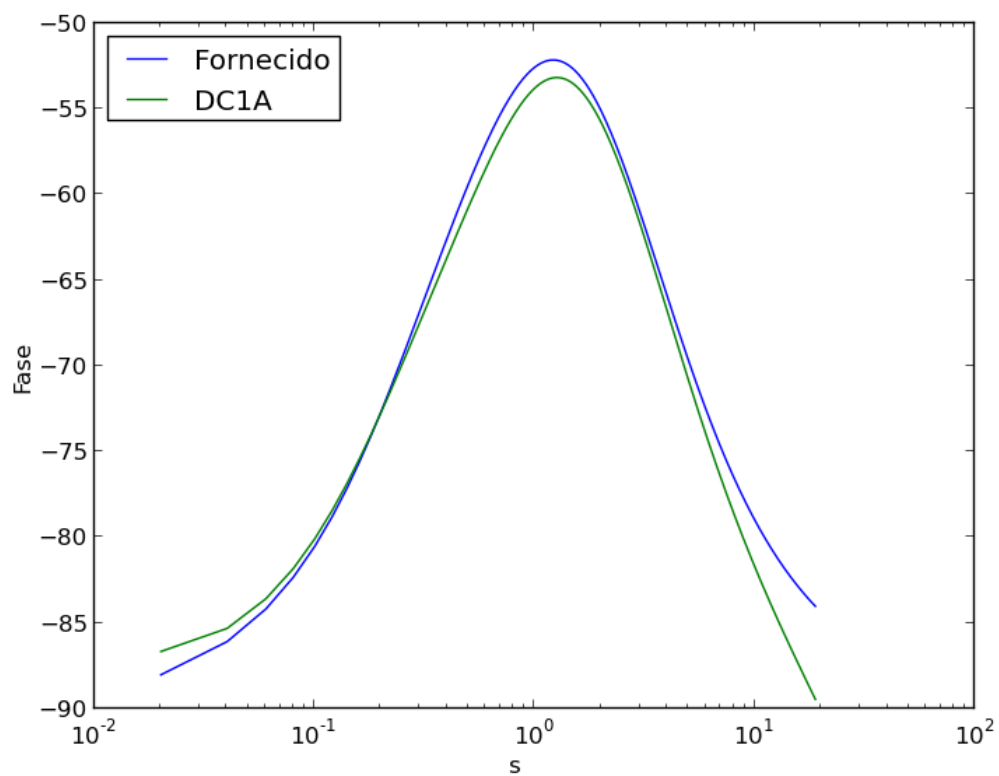


Figura 5.10 - Diagramas de bode - Fase (solução obtida para o caso de estudo 2)

A resposta temporal obtida utilizando o modelo 1 no simulink apresenta-se na figura 5.11.

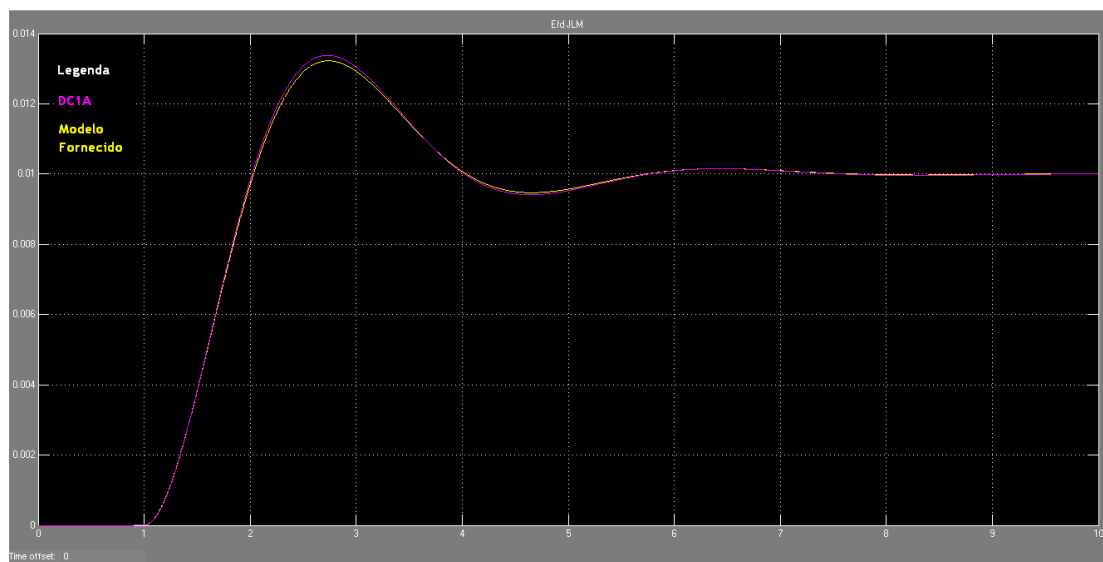


Figura 5.11 - Resposta temporal de V (solução obtida para o caso de estudo 2)

Com estes resultados, mais uma vez se confirma que a metodologia desenvolvida foi capaz de atingir uma boa solução.

5.2.4 - Caso de estudo 3

Após a execução do script para o modelo 1 com os valores do caso de estudo 2 (ver tabela 5.1) e uma inicialização dos parâmetros entre $[0,10]$, foi obtido o seguinte ajuste para os parâmetros do modelo DC1A.

Tabela 5.5 - Valor dos parâmetros para o modelo DC1A obtidos para o caso de estudo 3

Ka	8,52819675356945
Ke	0,05193713627559196
Kf	2,4638211792025557
Ta	0,0051760548075927785
Tb	3,95178519809685
Tc	6,244795170215725
Te	2,477422457856174
Tf	6,193762253680205

Este ajuste originou os diagramas de bode, para o módulo e a fase das respostas dos dois modelos no domínio das frequências, apresentados nas figuras 5.12 e 5.13.

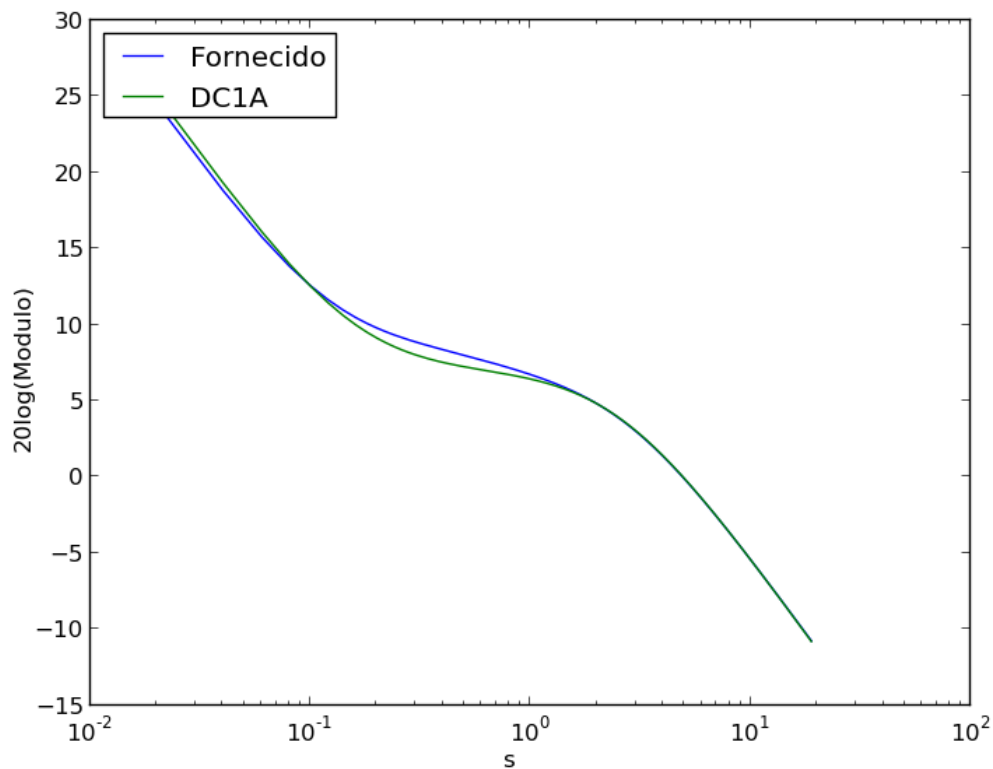


Figura 5.12 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 3)

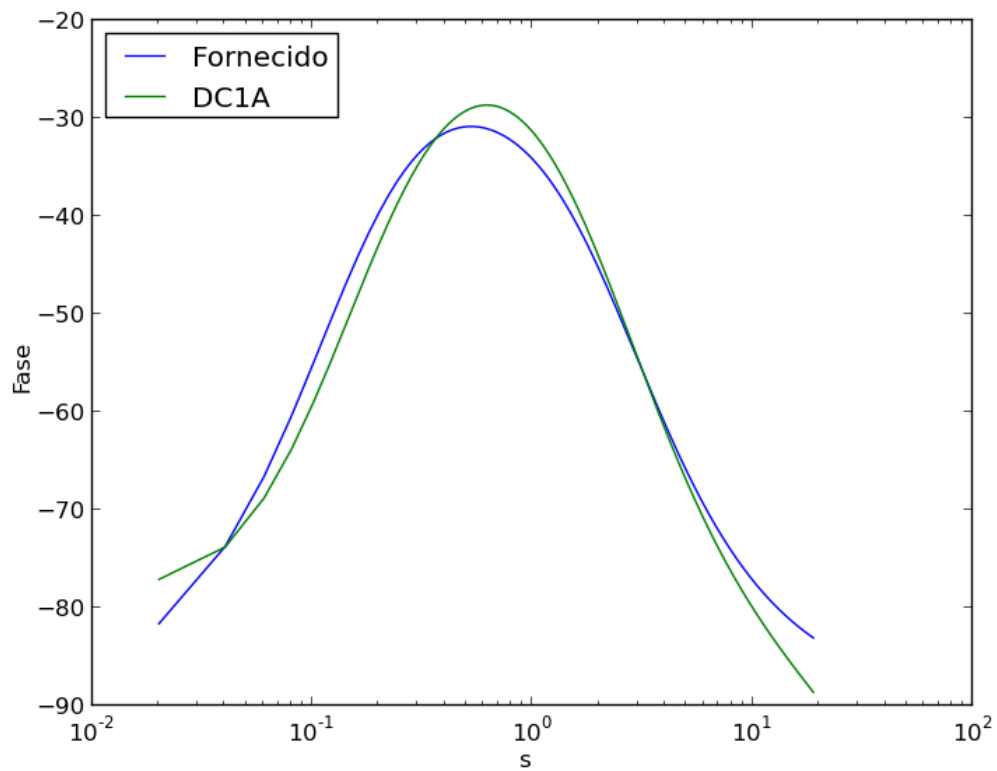


Figura 5.13 - Diagramas de bode - Fase (solução obtida para o caso de estudo 3)

A resposta temporal de V , obtida para um aumento em degrau de 0,01 p.u. em V_{ref} , apresenta-se na figura 5.14, a qual se confirma ser de boa qualidade.

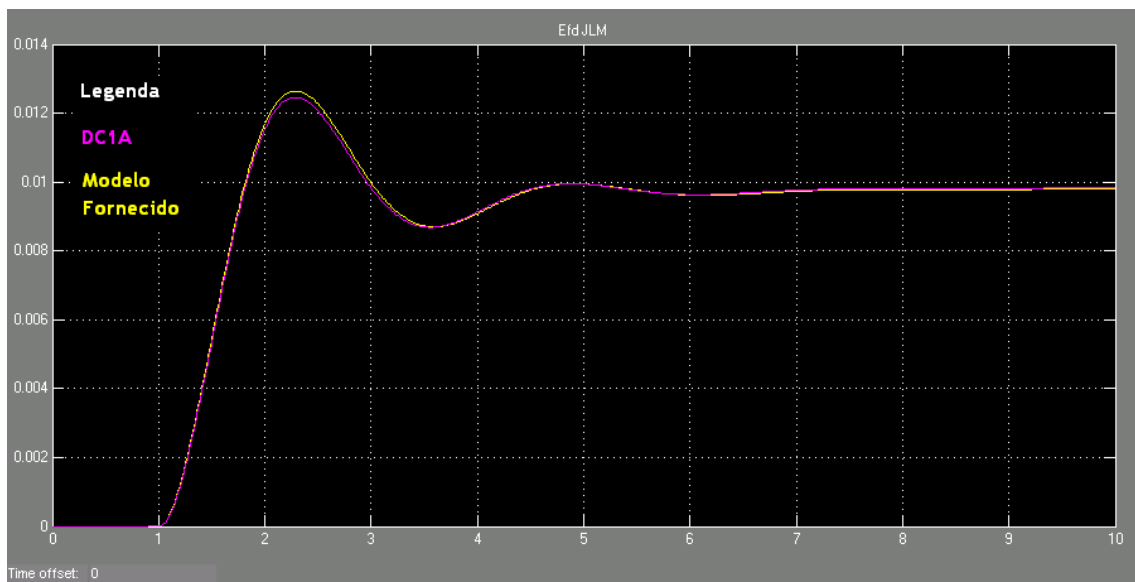


Figura 5.14 - Resposta temporal de V (solução obtida para o caso de estudo 3)

5.3 - Casos de estudo do modelo 2

Para o modelo 2, cujo diagrama de blocos se descreve na figura 5.2, foram fornecidos os seguintes casos de estudo:

Tabela 5.6 - Casos de estudo do modelo 2

Parâmetros do modelo 2	Caso 4	Caso 5	Caso 6	Caso 7
A	0	0	0	0
B	0,5915	1	0,7	0,7
C	3	1,82	2	9
D	2,3	1,3	4	4
E	0	0	0	0,1
F	0	0	0	0
G	0,1	2,1	0,1	0,1
H	2	3,7	4	4
I	1	1	1	1

5.3.1 - Caso de estudo 4

Após a execução do programa para o modelo 2 com os valores do caso de estudo 4 (ver tabela 5.6) e uma inicialização dos parâmetros entre [0,10], foi obtido o seguinte ajuste para os parâmetros do modelo DC1A.

Tabela 5.7 - Valor dos parâmetros do modelo DC1A obtidos para o caso de estudo 4

Ka	6,744612120640922
Ke	1,66944328361641e-05
Kf	0,974097038077393
Ta	0,01258476272590996
Tb	0,25013043736058793
Tc	0,07107466672957413
Te	8,755448355056856
Tf	1,443507050917621

Este ajuste originou os diagramas de bode que se apresentam na figura 5.15 e 5.16.

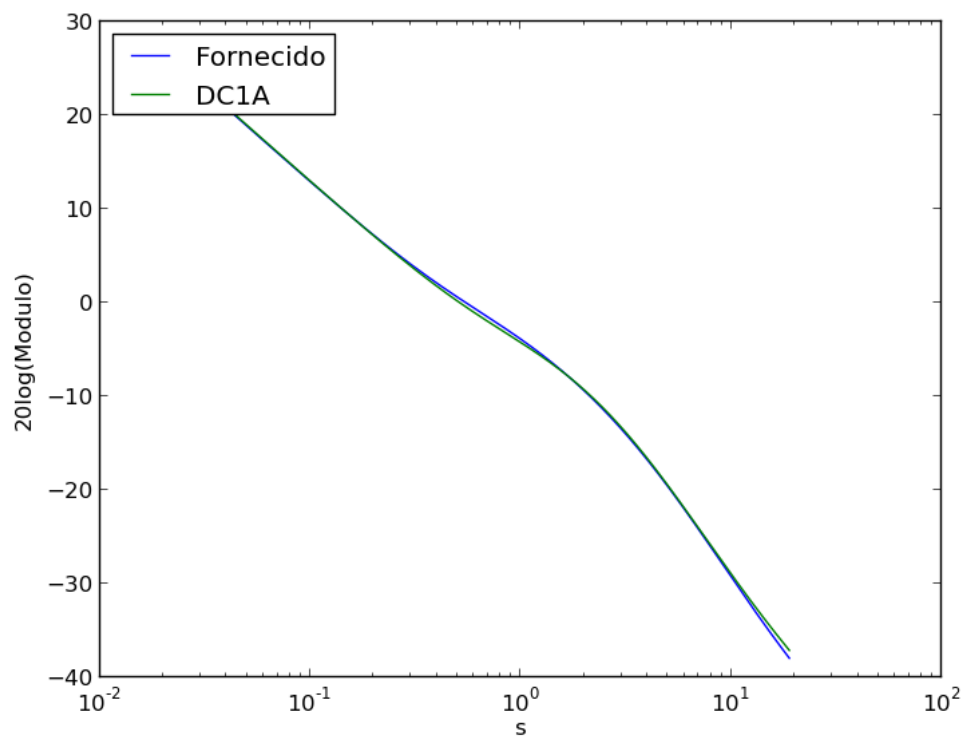


Figura 5.15 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 4)

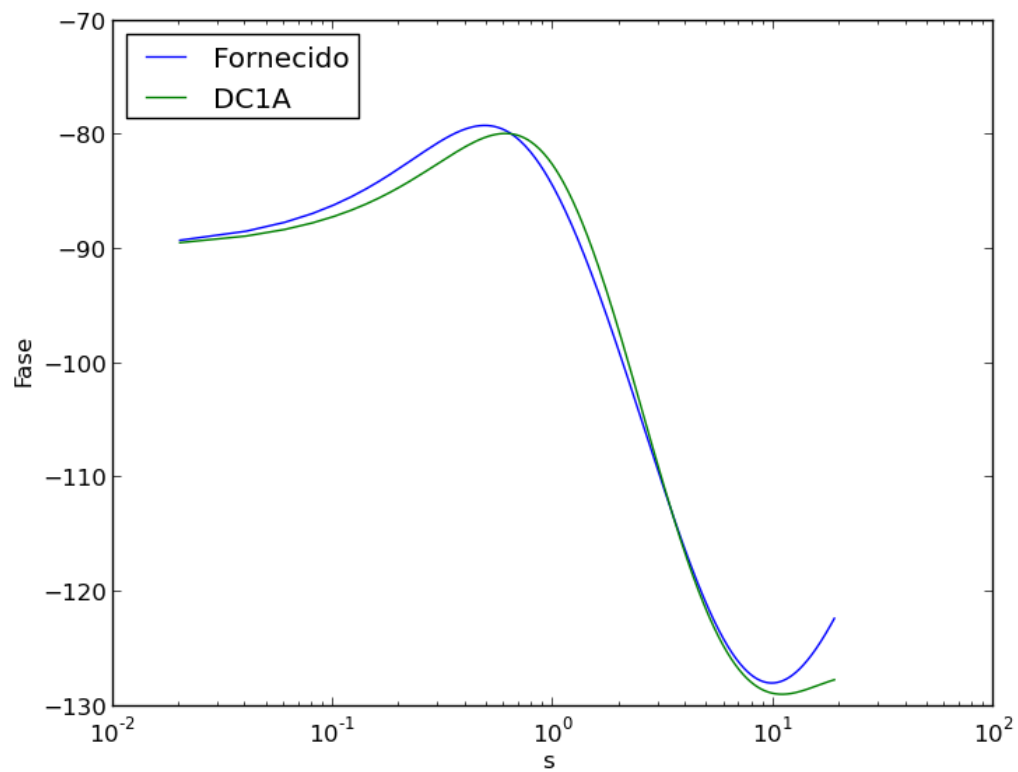


Figura 5.16 - Diagramas de bode - Fase (solução obtida para o caso de estudo 4)

A resposta temporal obtida, para a perturbação até agora considerada, apresenta-se na figura 5.17.

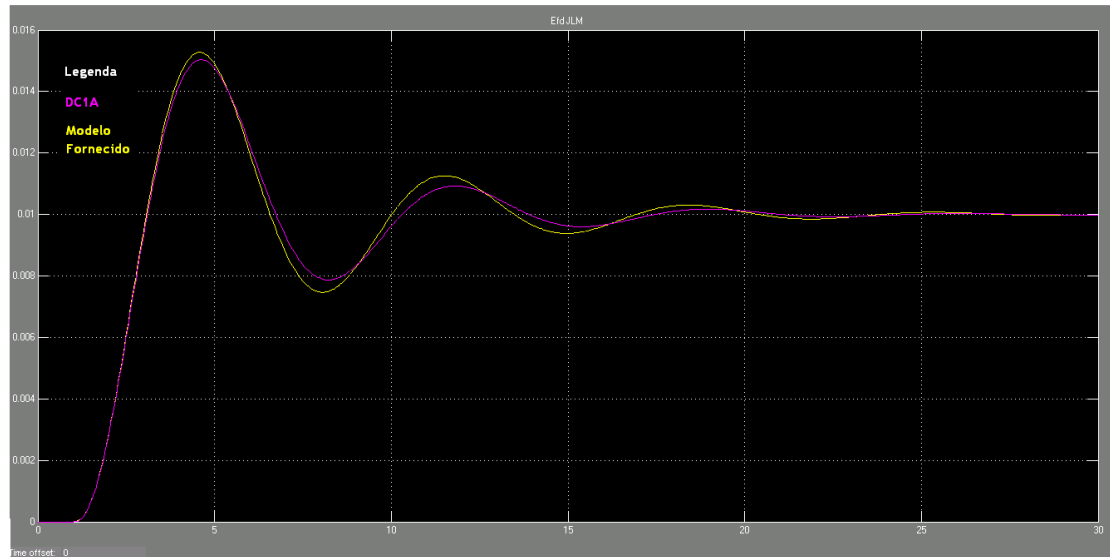


Figura 5.17 - Resposta temporal de V (solução obtida para o caso de estudo 4)

Os resultados obtidos observados da figura 5.15 até à figura 5.17 demonstram que a metodologia desenvolvida conseguiu encontrar uma solução com qualidade para este caso de estudo.

5.3.2 - Caso de estudo 5

Após a execução do script para o modelo 2 com os valores do caso de estudo 5 (ver tabela 5.6) e uma inicialização dos parâmetros entre $[0,10]$, foi obtido o seguinte ajuste para os parâmetros do modelo DC1A.

Tabela 5.8 - Valor dos parâmetros do modelo DC1A obtidos para o caso de estudo 5

Ka	10,291646467451052
Ke	1e-05
Kf	0,9289758312554259
Ta	4,537682143557354
Tb	0,005676176391016623
Tc	3,644070305416143
Te	3,834892626872191
Tf	3,8608255851040116

Este ajuste originou os diagramas de bode que se apresentam na figura 5.18 e 5.19.

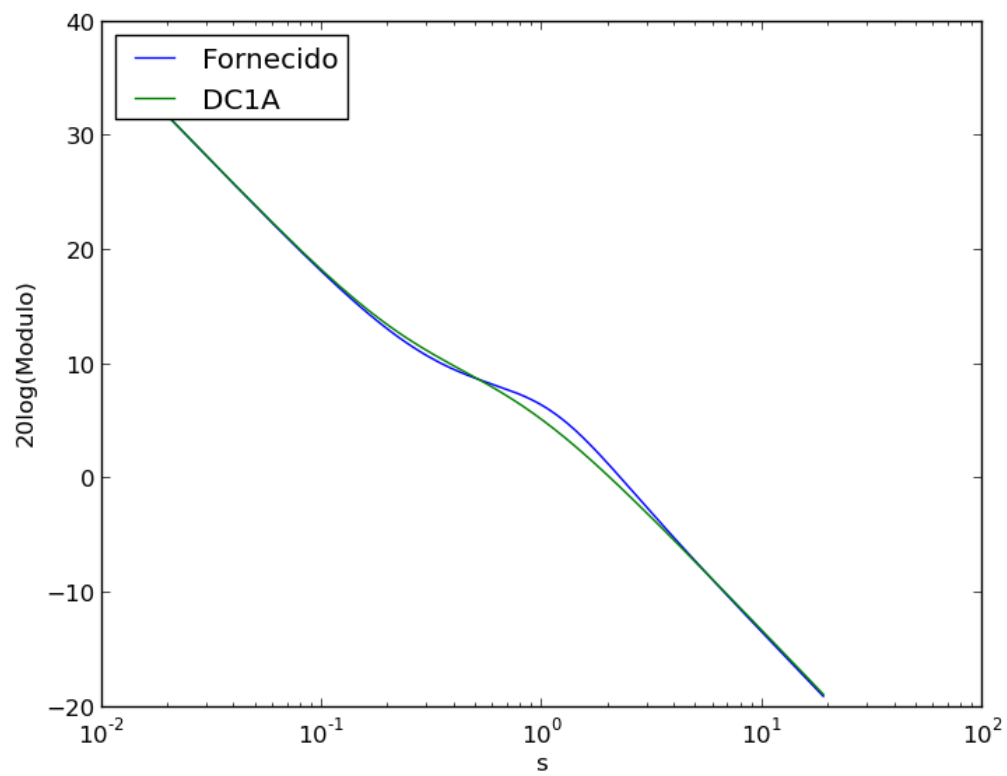


Figura 5.18 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 5)

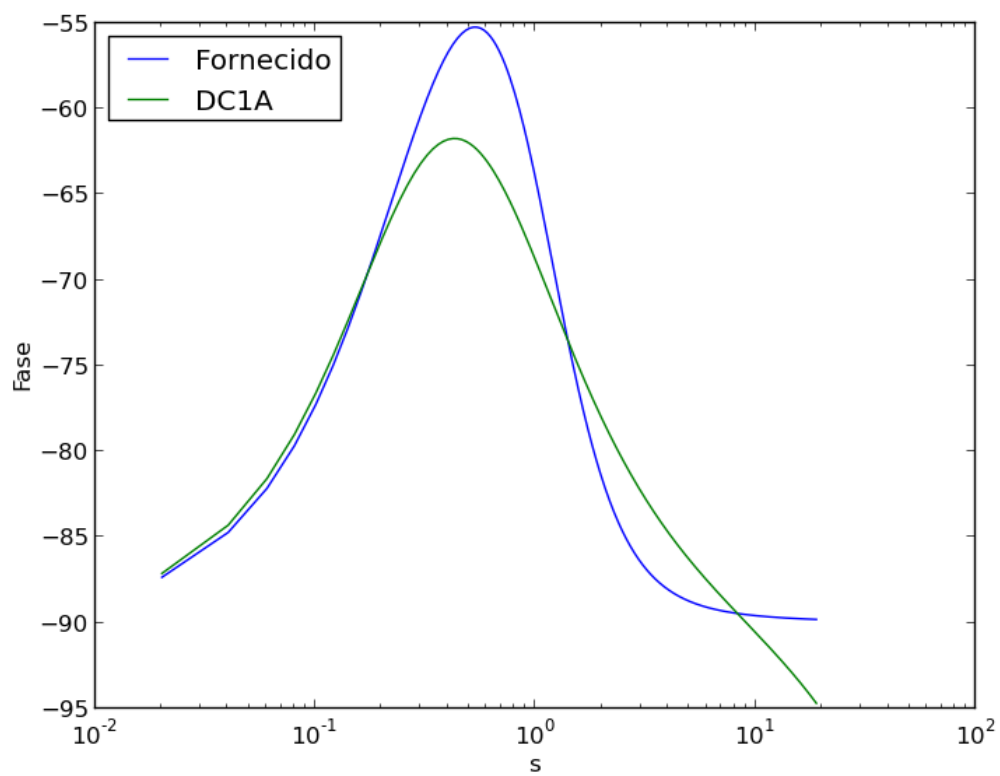


Figura 5.19 - Diagramas de bode - Fase (solução obtida para o caso de estudo 5)

A resposta temporal obtida, para a perturbação até agora considerada, apresenta-se na figura 5.20.

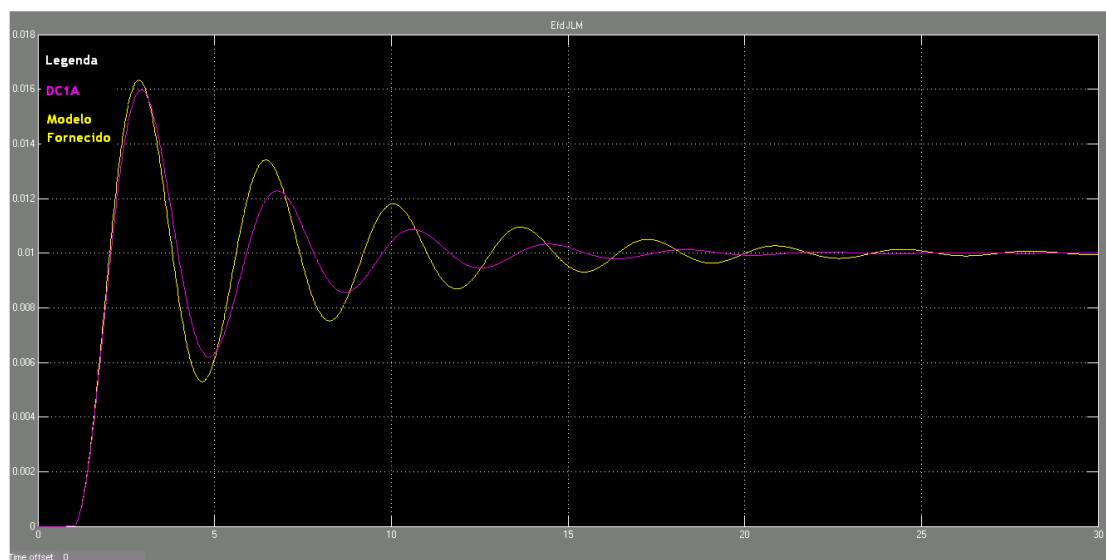


Figura 5.20 - Resposta temporal de V (solução obtida para o caso de estudo 5)

Neste caso o script não foi capaz de fazer um ajuste tão bom como nos casos de estudo anteriores. Isto pode dever-se ao facto de as boas soluções não estarem próximas da solução inicial dos parâmetros entre 0 e 10. Correu-se então o script para valores iniciais típicos dos parâmetros do modelo DC1A, retirados de [12], obtendo-se os seguintes resultados.

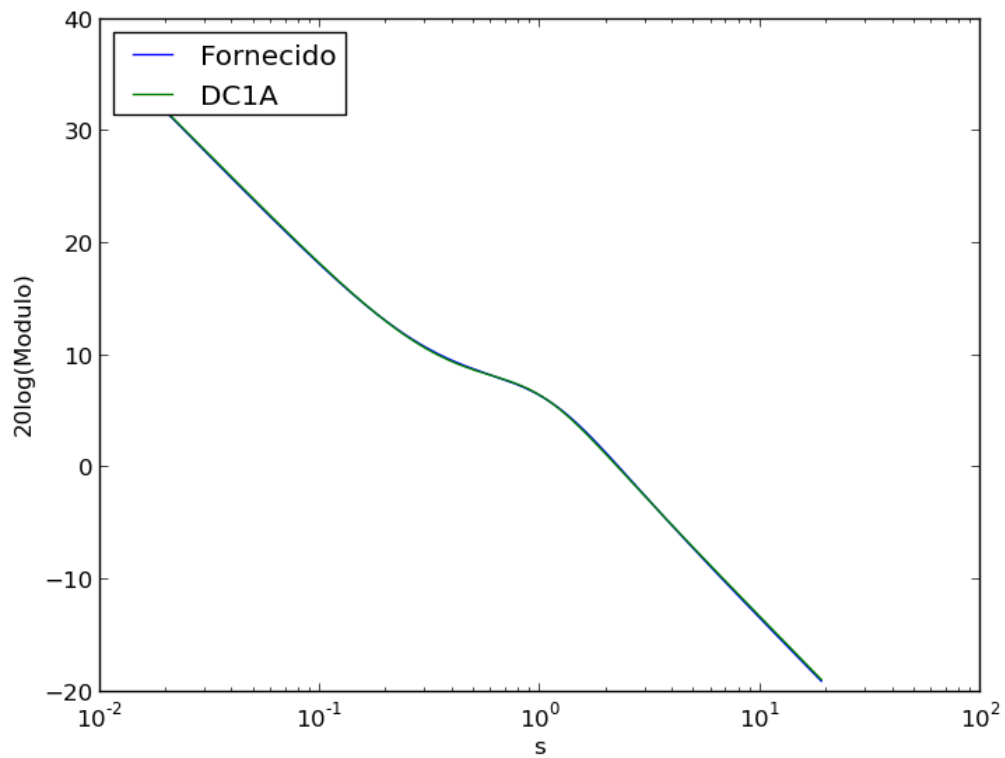


Figura 5.21 - Diagramas de bode - Módulo (solução obtida para a nova solução do caso de estudo 5)

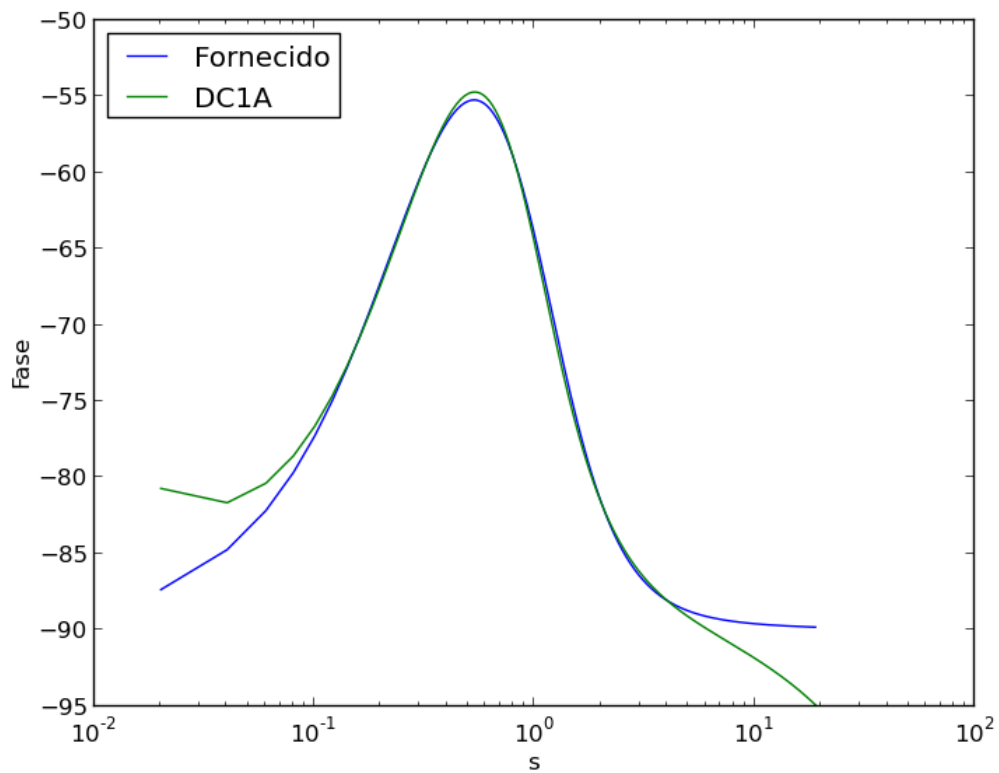


Figura 5.22 - Diagramas de bode - Fase (solução obtida para a nova solução do caso de estudo 5)

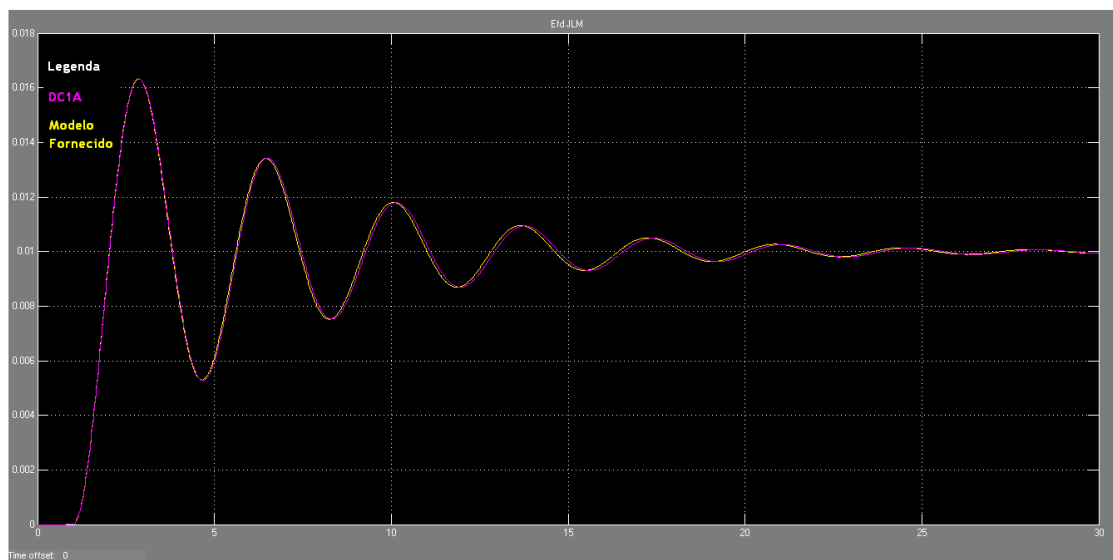


Figura 5.23 - Resposta temporal de V (solução obtida para a nova solução do caso de estudo 5)

Destes resultados se observa que a nova solução encontrada é muito melhor do que a obtida sem grande manipulação da solução inicial, o que realça a importância da solução inicial para a convergência do EPSO.

5.3.3 - Caso de estudo 6

Após a execução do script para o modelo 2 com os valores do caso de estudo 6 (ver tabela 5.6) e uma inicialização dos parâmetros entre $[0,10]$, foi obtido o seguinte ajuste para os parâmetros do modelo DC1A.

Tabela 5.9 - Valor dos parâmetros do modelo DC1A obtidos para o caso de estudo 6

Ka	10,73340518092439
Ke	0,0990671418588721
Kf	3,6456989565267044
Ta	19,74596648344778
Tb	6,6951824659710315
Tc	2,4657734202759514
Te	0,03273504495054162
Tf	3,732897454106783

Este ajuste originou os diagramas de bode que se apresentam na figura 5.24 e 5.25.

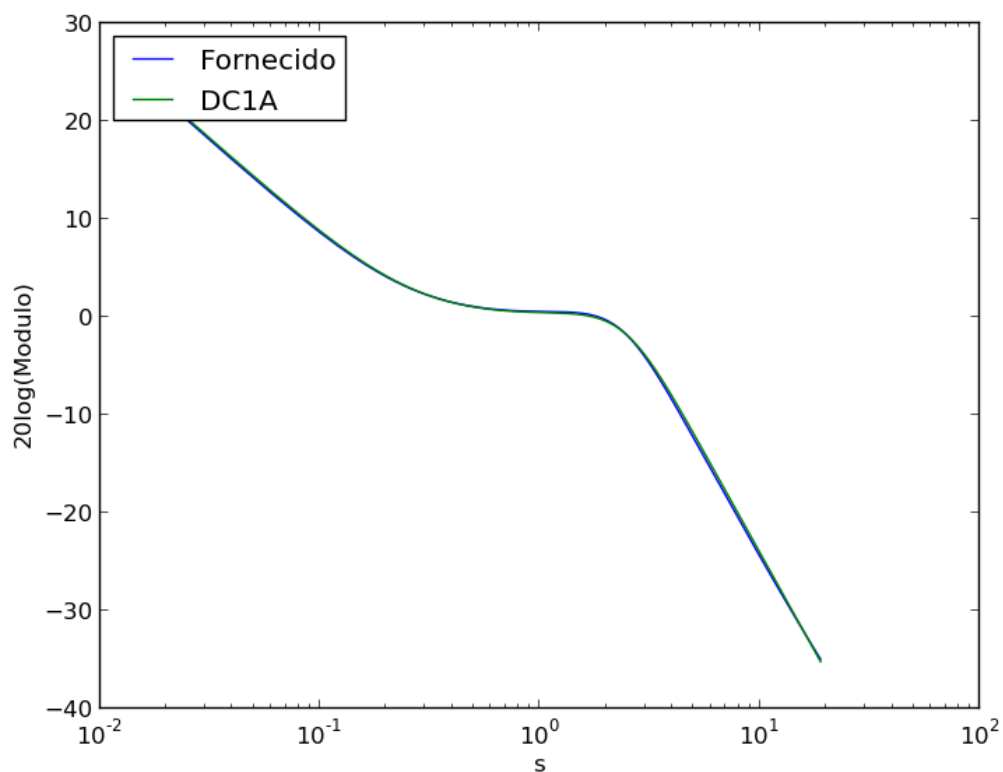


Figura 5.24 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 6)

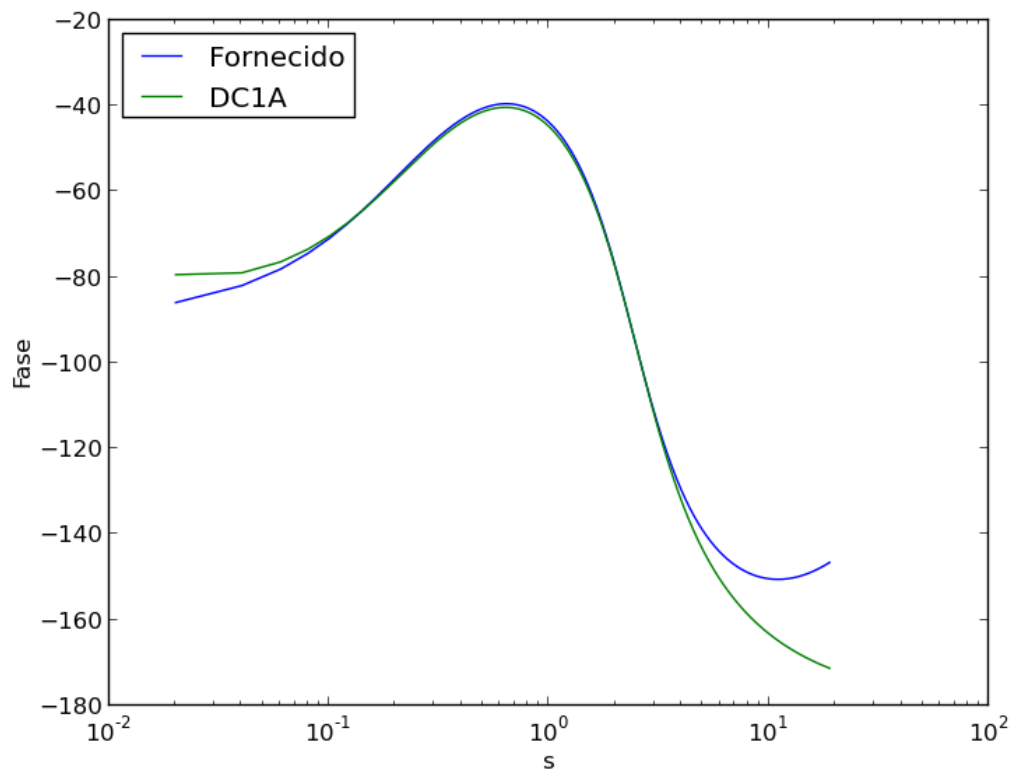


Figura 5.25 - Diagramas de bode - Fase (solução obtida para o caso de estudo 6)

A resposta temporal obtida, para a perturbação até agora considerada, apresenta-se na figura 5.26.

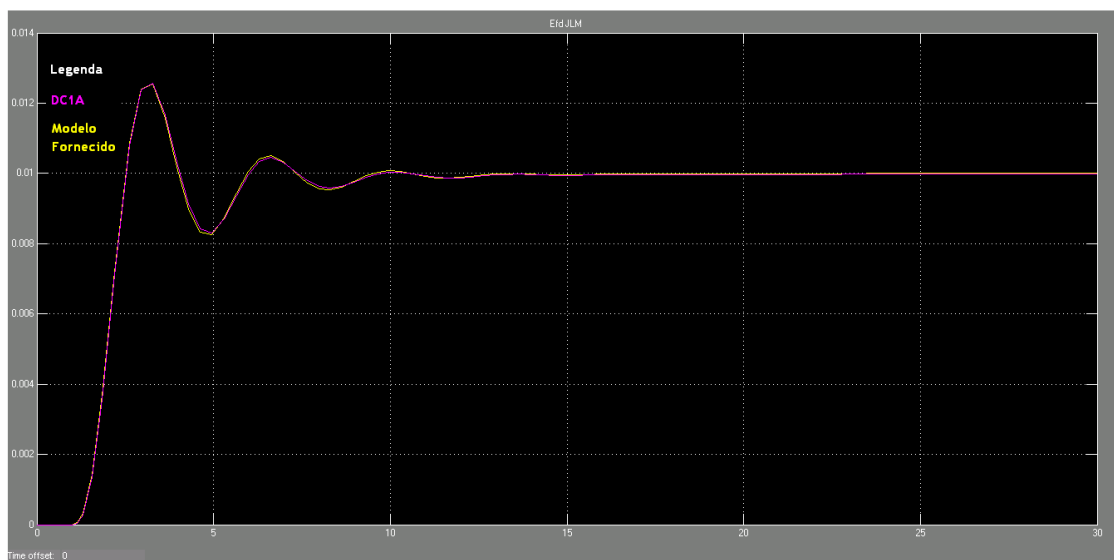


Figura 5.26 - Resposta temporal de V (solução obtida para o caso de estudo 6)

Os resultados obtidos observados da figura 5.24 até à figura 5.26 demonstram que a metodologia desenvolvida conseguiu encontrar uma solução com qualidade para este caso de estudo.

5.3.4 - Caso de estudo 7

Após a execução do script para o modelo 2 com os valores do caso de estudo 7 (ver tabela 5.6) e uma inicialização dos parâmetros entre $[0,10]$, foi obtido o seguinte ajuste para os parâmetros do modelo DC1A.

Tabela 5.10 - Valor dos parâmetros do modelo DC1A obtidos para o caso de estudo 7

Ka	19,406039251675924
Ke	1,89752619220438
Kf	2,1512012516823082
Ta	4,795427185099345
Tb	12,907294558369575
Tc	2,6924977830694097
Te	0,09877980900385379
Tf	3,5269100864163376

Este ajuste originou os diagramas de bode que se apresentam na figura 5.27 e 5.28.

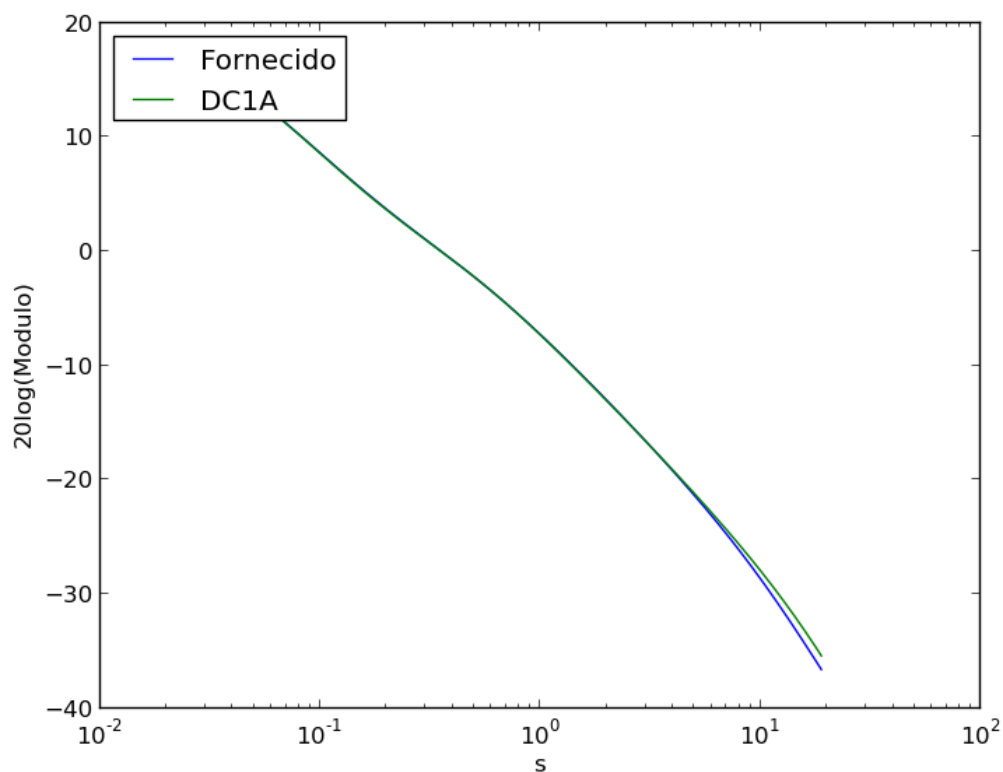


Figura 5.27 - Diagramas de bode - Módulo (solução obtida para o caso de estudo 7)

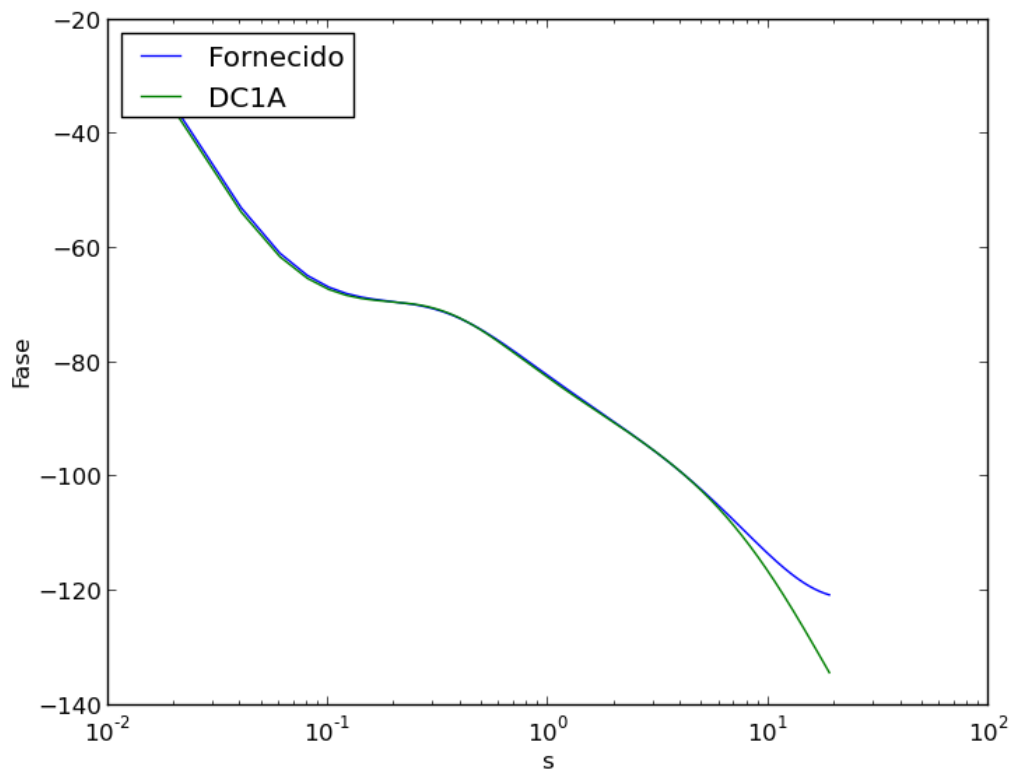


Figura 5.28 - Diagramas de bode - Fase (solução obtida para o caso de estudo 7)

A resposta temporal obtida, para a perturbação até agora considerada, apresenta-se na figura 5.29.

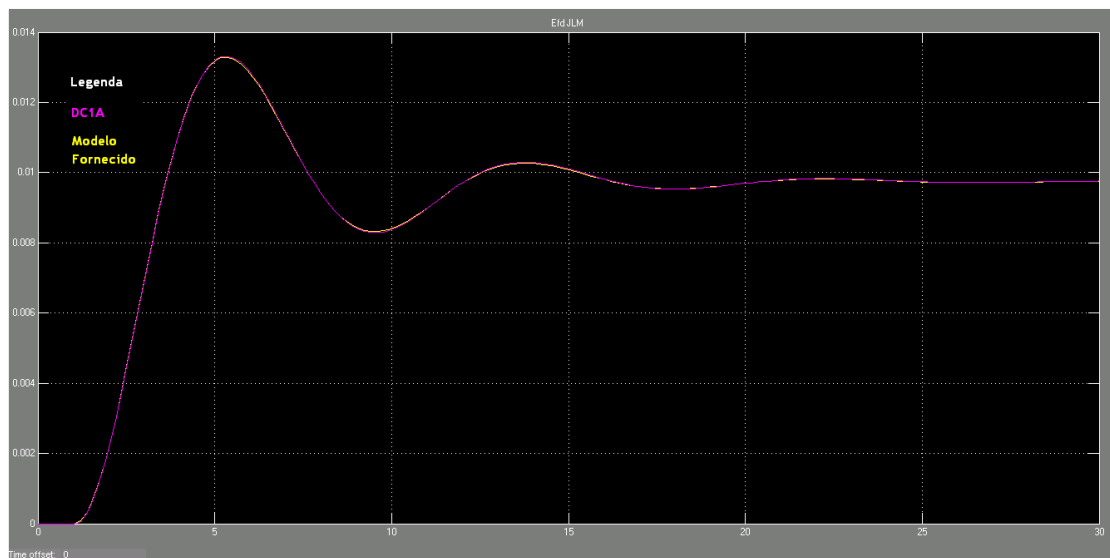


Figura 5.29 - Resposta temporal de V (solução obtida para o caso de estudo 7)

Os resultados obtidos observados da figura 5.24 até à figura 5.26 demonstram que a metodologia desenvolvida conseguiu encontrar uma solução com qualidade para este caso de estudo.

5.4 - Adaptação do modelo 1 ao modelo 2 dos casos de estudo

Com o intuito de testar ainda mais a capacidade de ajuste da aplicação desenvolvida, decidiu-se utilizar o modelo fornecido 1, dos casos de estudo anteriores, para se adaptar ao modelo fornecido 2, tendo o modelo 2 o valor de parâmetros correspondente ao caso de estudo 4. O resultado obtido para o ajuste de parâmetros do modelo 1 foi o seguinte:

Tabela 5.11 - Parâmetros obtidos para o ajuste do modelo fornecido 1

Ta1	4,63574370045
Ta11	2,29827336158
Ta2	5,01289019398
Ta22	3,99790242252
Tee	4,41895029953

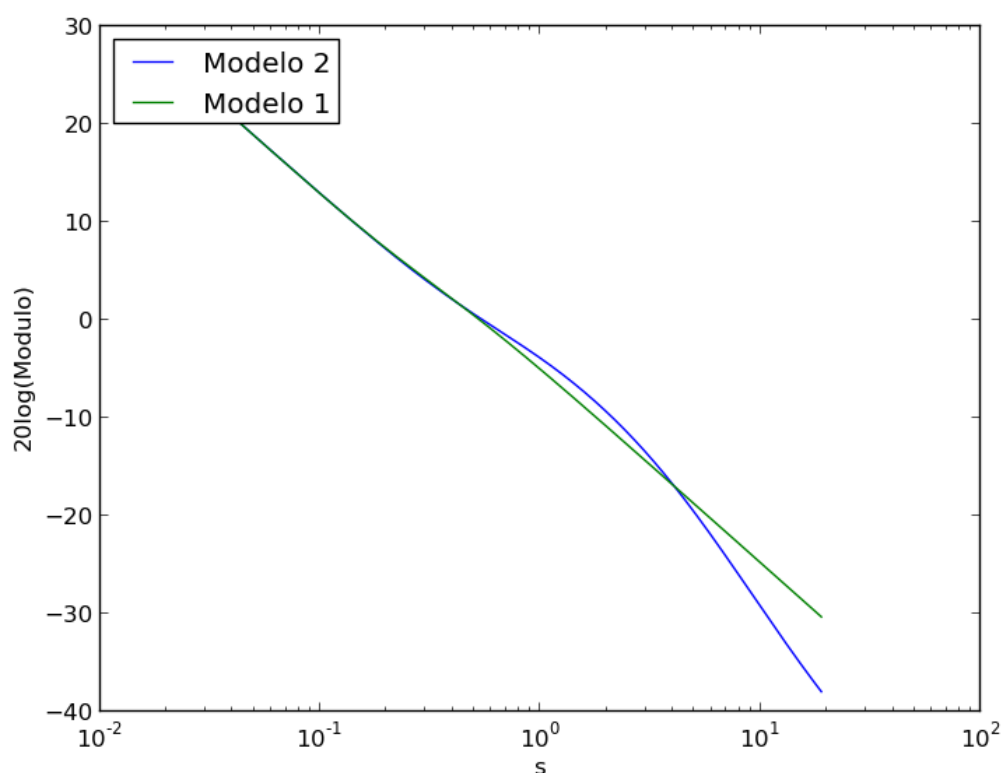


Figura 5.30 - Diagramas de bode - Módulo (solução obtida para a adaptação do modelo fornecido 1 ao caso de estudo 4)

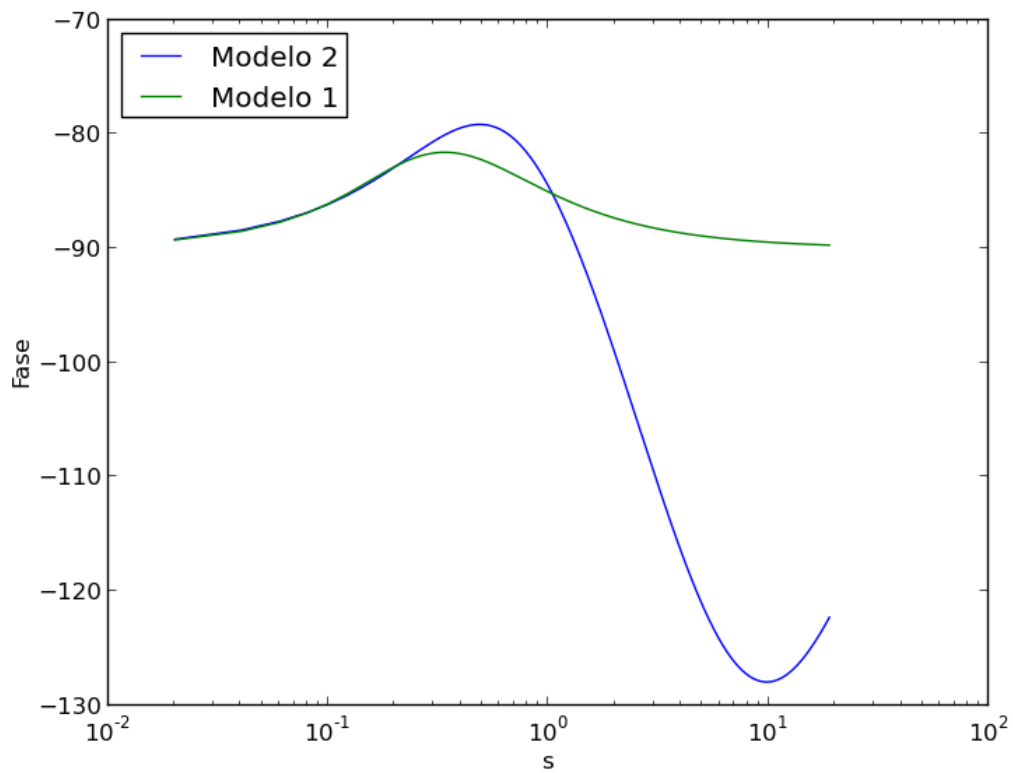


Figura 5.31 - Diagramas de bode - Fase (solução obtida para a adaptação do modelo fornecido 1 ao caso de estudo 4)

Os resultados obtidos na figura 5.31 levam a concluir que aparentemente as funções não são muito compatíveis, mas ao observar o gráfico tridimensional da figura 5.32 é possível verificar que, efetivamente o algoritmo está muito próximo de conseguir um ajuste perfeito.

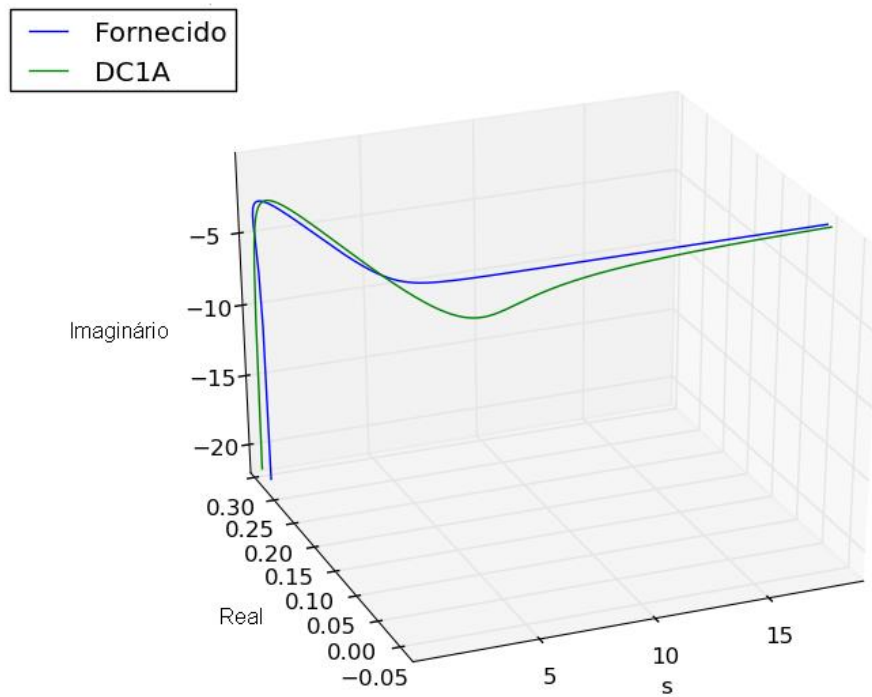


Figura 5.32 - Gráfico tridimensional da adaptação do modelo 1 ao caso de estudo 4

A resposta temporal obtida, para a perturbação até agora considerada, apresenta-se na figura 5.33.

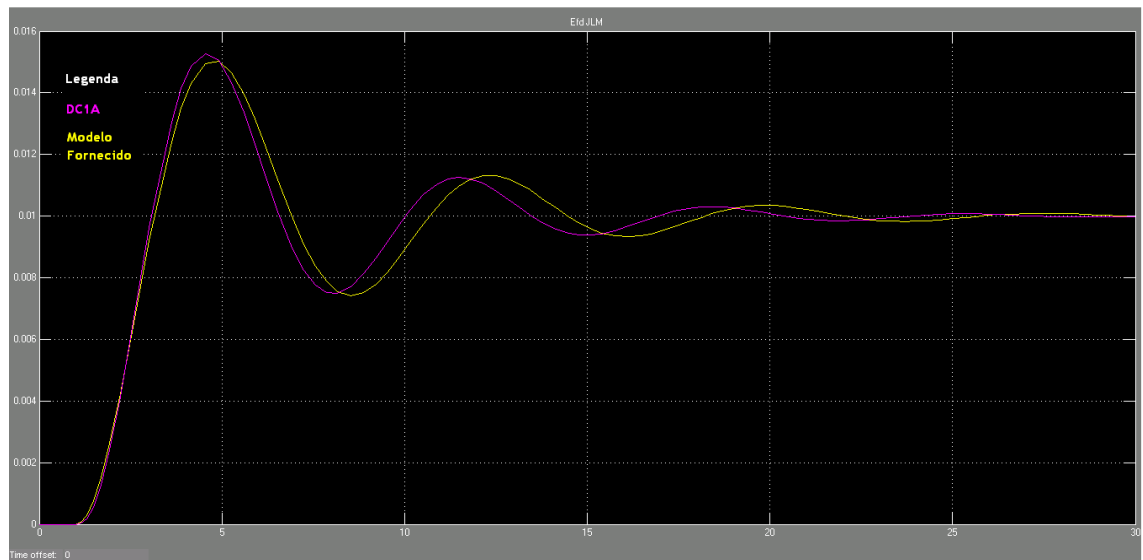


Figura 5.33 - Resposta temporal de V (solução obtida para o caso de estudo 7)

Capítulo 6

Conclusões

Em conclusão, nesta dissertação foi desenvolvida uma aplicação computacional baseada no algoritmo EPSO para efetuar o ajuste do modelo DC1A (ou outros) a modelos de sistemas de excitação fornecidos e respetivo manual de utilização. Essa funcionalidade foi testada para dois modelos fornecidos e 8 casos de estudo. Foi também desenvolvido um novo método de evitar o colapso do enxame que pode ser aplicado tanto em algoritmos PSO como em algoritmos do tipo EPSO. A partir dos resultados obtidos para os casos de estudo, chegou-se às seguintes conclusões:

- A meta-heurística EPSO desenvolvida é adequada para o ajuste dos parâmetros do modelo DC1A.
- A inicialização do enxame é muito importante para a convergência do algoritmo de forma eficaz pelo que deve ser garantido, sempre que possível, uma inicialização das partículas aleatoriamente, por todo o domínio dos seus parâmetros.
- O modelo DC1A consegue adaptar-se adequadamente a todos os modelos testados como casos de estudo.
- A aplicação desenvolvida tem flexibilidade de utilização suficiente para que o seu utilizador possa inserir uma qualquer função de transferência, quer para o modelo a ajustar, quer para o modelo fornecido a ser utilizado como referência de ajuste.

Devido à qualidade dos resultados obtidos com a metodologia desenvolvida, é expectável que o algoritmo possa ser aplicado futuramente a modelos dinâmicos de outro tipo de dispositivos.

Os resultados produzidos pelo EPSO para o tipo de função objetivo utilizada neste trabalho tornam interessante a aplicação do algoritmo a outros problemas em que uma função objetivo semelhante possa ser aplicada. Um exemplo e sugestão seriam aplicar o EPSO na resolução das equações diferenciais “Navier-Stokes” da hidrodinâmica [15], tomando como função objetivo o módulo da diferença entre os dois lados da equação.

Referências

- [1] IEEE Power Engineering Society (2006). “IEEE Recommended Practice for Excitation System Models for Power System Stability Studies”, IEEE Std. 421.5-2005.
- [2] Vasconcelos, H. (2013) Apontamentos da unidade curricular DESI do 5º ano do MIEEC da área de energia, ano lectivo 2012/13.
- [3] Marques, V. (2011). Ajuste de parâmetros para modelos típicos de sistemas de excitação, recorrendo à resposta em frequência do modelo. Dissertação de Mestrado FEUP, 2011.
- [4] Fogel, D. (2006) Evolutionary Computation: Toward a New Philosophy of Machine Intelligence (3rd ed.). Piscataway, NJ: IEEE Press.
- [5] Holland, J. (1992). Adaptation in Natural and Artificial Systems. Cambridge, MA: MIT Press.
- [6] Rechenberg, I. (1994). Evolutionsstrategie '94, Stuttgart: Fromman-Holzboog.
- [7] Schwefel, Hans-Paul (1974): Numerische Optimierung von Computer-Modellen (PhD thesis). Reprinted by Birkhäuser (1977).
- [8] Kennedy, J.; Eberhart, R. (1995). "Particle Swarm Optimization". *Proceedings of IEEE International Conference on Neural Networks IV*.
- [9] Rechenberg I (1971) Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution. Dr.-Ing. Thesis, Technical University of Berlin, Department of Process Engineering.
- [10] Carvalho, J. (2000); Sistemas de controle automático. LTC Editora, Rio de Janeiro.
- [11] Fogel, L. J.; A.J. Owens, A. J.; Walsh (1964) “On the evolution of artificial intelligence,” Proc. 5th National Symp. On Human Factors in Engineering, IEEE, San Diego, CA, pp. 63-76.
- [12] Siemens Energy, Inc (2009). PSSE 32.0 Online Documentation. Acedido em <http://www.sourcecodeprojects.com/417601/>
- [13] Miranda, V. ; Fonseca, N. (2002). “EPSO - Best-Of-Two-Worlds Meta-Heuristic Applied To Power System Problems”. Evolutionary Computation, 2002. CEC '02.
- [14] Prabha Kundur, “Power System Stability and Control”, McGraw-Hill, 1993.
- [15] C.B. Parker, “Encyclopedia of Physics (2nd edition)”, McGraw-Hill, 1994.

Anexos

Manual de utilização do algoritmo EPSO para ajuste do modelo DC1A a outros modelos

João Luís Rodrigues de Castro Mendes Martins

20/5/2013

Resumo

Este documento tem como objetivo facilitar a utilização do script de ajuste do modelo DC1A para reguladores de tensão a outros modelos, para utilizadores com poucos conhecimentos de python ou de programação. Nele são retratados problemas típicos e soluções passo a passo, com exemplos.

Conteúdo

0.1	Introdução	68
0.1.1	Requisitos de software para correr o script	68
0.1.2	Instruções gerais	68
0.2	Alteração do modelo fornecido	69
0.3	Alteração do modelo a ajustar	69
0.3.1	Geral	69
0.3.2	Solução inicial	71
0.3.3	Limites mínimos dos parâmetros	71
0.4	Número máximo de iterações/Número de partículas:	71
0.4.1	Convergência	71
0.4.2	Controlar as seeds aleatórias	73
0.5	Ativar e desativar gráficos auxiliares	73
0.6	Ficheiros de saída	73
0.6.1	Modificar a pasta onde vão ser gravados os ficheiros de saída	

0.1 Introdução

0.1.1 Requisitos de software para correr o script

Para correr o script é necessária uma GUI (Guided User Interface) de python, e ter as bibliotecas “Numpy” e “Matplotlib”, que podem ser descarregadas no site da dissertação (paginas.fe.up.pt/~ee08020) em “links”.

0.1.2 Instruções gerais

Ao utilizar o script, todas as zonas a serem modificadas estão apresentadas da seguinte forma:

```
#inicio
...código a modificar
#fim
```

Não deve modificar fora dessas zonas, a não ser que saiba o que está a fazer. Cada uma dessas zonas possui também uma frase chave, mais ou menos centrada no script, pelo que deve sempre verificar que está a modificar no sítio certo, verificando que a frase chave é a correta:

```
#Modelo fornecido
#inicio
Ta1=3
Ta11=1.3
Ta2=0.7
Ta22=0.7
Tee=0.65
#fim
```

Neste caso a frase chave é a frase “Modelo fornecido”.

Deve certificar-se também que a indentação das suas modificações é igual à indentação utilizada anteriormente:

```
#inicio
a modificar 1
    a modificar 2
#fim
```

deve resultar nas seguintes modificações:

```
#inicio
modificação 1
    a modificação 2
#fim
```

0.2 Alteração do modelo fornecido

Para alterar o modelo fornecido, deve procurar pela frase chave “Modelo fornecido” e aí, deve nomear, e atribuir valores às variáveis do modelo que vai ser fixo. Note que é necessário o nome das variáveis comecem com uma letra maiúscula para não interferir com as variáveis do sistema. Exemplo (utilizado para os casos de estudo 4-8):

```
#Modelo fornecido
#inicio
A=0
B=0.5915
C=3
D=2.3
E=0
F=0
G=0.1
H=2
I=1
#fim
```

Em seguida, deve procurar pela frase chave “Resposta do modelo fornecido”, identificar a função `R.append()`, apagar o que está entre parênteses, e substituir pela expressão desejada, utilizando os nomes das variáveis já utilizadas e “s” a representar $s = \sigma + j\omega$. Importante: $**=^$, ex: $2**3=8=2*2*2$. Exemplo (utilizado para os casos de estudo 4-8):

```
R.append((F*s**3+G*s**2+H*s+I)/(A*s**4+B*s**3+C*s**2+D*s+E))
```

que corresponde a $\frac{F.s^3+G.s^2+H.s+I}{A.s^4+B.s^3+C.s^2+D.s+E}$.

Podemos também fazer cálculos auxiliares antes de chamar a função `append` (utilizado nos casos 1-3):

```
N=(s*Ta1+1)*(s*Ta2+1)
D1=(s*Ta22)*(1+s*Tee)
D=s*Ta11*(D1+1+s*Ta2)
R.append(N/D)
```

É de notar, no entanto, que as variáveis auxiliares têm todas a primeira letra maiúscula, para não interferir com as outras variáveis no programa.

0.3 Alteração do modelo a ajustar

0.3.1 Geral

Para alterar o modelo a ajustar, deve procurar pela frase chave “Parametros DC1A”, e colocar na variável “parametros” o número de variáveis que a expressão do modelo a ajustar vai ter. Exemplo:

```
#inicio
parametros=8
#fim
```

De seguida, para esse numero de parametros é necessário criar uma legenda, reverente a um vetor “par”, criando assim para j parametros uma linha de comentário par[i][j] = Nome do atributo j. É de notar que a legenda, assim como a frase chave, encontra-se comentada a verde. Exemplo:

```
par[i][0]=Tc
par[i][1]=Tb
par[i][2]=Te
par[i][3]=Ka
par[i][4]=Ke
par[i][5]=Kf
par[i][6]=Tf
par[i][7]=Ta
```

Após a legenda a verde, é necessário também atribuir esses nomes por ordem ao vetor “ordem”, utilizando sempre o nome entre ‘plicas’:

```
ordem=['Tc','Tb','Te','Ka','Ke','Kf','Tf','Ta']
```

Por fim, é necessário procurar pela frase chave “Modelo a ajustar”, atribuir ao vetor Rpar o valor da expressão. Neste caso não é possível utilizar o nome das variáveis, pelo que vai ser necessário usar a legenda para identificar qual é o seu nome na particula “par”, e utilizando f[j] como $s = \sigma + j\omega$ (expressão do DC1A):

```
Rpar[i][j]=(((f[j]*par[i][0]+1)/(f[j]*par[i][1]+1))*
((par[i][3])/(f[j]*par[i][7]+1))*((1)/(f[j]*par[i][2]+
par[i][4])))/(1+((f[j]*par[i][5])/(f[j]*par[i][6]+1))*
((f[j]*par[i][0]+1)/(f[j]*par[i][1]+1))*((par[i][3])/
(f[j]*par[i][7]+1))*((1)/(f[j]*par[i][2]+par[i][4]))))
```

Que é equivalente a:

$$\frac{\frac{f[j].par[i][0]+1}{f[j].par[i][1]+1} \times \frac{par[i][3]}{f[j].par[i][7]+1} \times \frac{1}{f[j].par[i][2]+par[i][4]}}{1 + \frac{f[j].par[i][5]}{f[j].par[i][6]+1} \times \frac{f[j].par[i][0]+1}{f[j].par[i][1]+1} \times \frac{par[i][3]}{f[j].par[i][7]+1} \times \frac{1}{f[j].par[i][2]+par[i][4]}}$$

A mesma função deverá ser armazenada em RbestG, perto da frase chave “Resposta do melhor global”, mas em vez de usar “par[i][j]” para um parâmetro, deve-se utilizar bestglobal[j]. Exemplo:

```
RbestG.append((((f[j]*bestglobal[0]+1)/
(f[j]*bestglobal[1]+1))*((bestglobal[3])/
(f[j]*bestglobal[7]+1))*((1)/(f[j]*bestglobal[2]+bestglobal[4])))/
(1+((f[j]*bestglobal[5])/(f[j]*bestglobal[6]+1))*
((f[j]*bestglobal[0]+1)/(f[j]*bestglobal[1]+1))*
((bestglobal[3])/(f[j]*bestglobal[7]+1))*
((1)/(f[j]*bestglobal[2]+bestglobal[4])))))
```

0.3.2 Solução inicial

Para uma maior eficácia do EPSO, a solução inicial deve ser gerada aleatoriamente no domínio dos parâmetros. A modificação da solução inicial pode ser feita a seguir à frase chave “Solucao inicial” e deve-se colocar em solinimax e solinimin para cada parametro o seu limite máximo e mínimo respetivamente. Exemplo para 8 parâmetros:

```
solinimax=[10,10,10,10,10,10,10,10]  
solinimin=[0,0,0,0,0,0,0,0]
```

0.3.3 Limites mínimos dos parâmetros

Devido ao significado físico dos parâmetros utilizados e para não ocorrer instabilidade da solução no domínio dos tempos, todos os parâmetros no EPSO são positivos. Na ocorrência de um parâmetro possuir outro limite que não *Parâmetro* ≥ 0 , isso deve ser modificado manualmente pelo utilizador a seguir à frase chave “Limites dos parametros” colocando no if os parametros e os seus limites. Os parametros devem ser referidos como par[i][numero do parâmetro na legenda], terminar com “:”, e como sempre deve ser respeitada a anterior identificação no ficheiro. Exemplo:

```
if par[i][1]<=0.005 or par[i][7]<=0.005 or par[i][2]<=0.005 or  
par[i][6]<=0.005:
```

0.4 Número máximo de iterações/Número de partículas:

Os parametros do EPSO a serem modificados podem ser encontrados a seguir à frase chave “Iteracoes e particulas”. Deve-se modificar o itermax para o número máximo de iterações pretendido, e o npar para o número de partículas pretendido. Exemplo:

```
itermax=400  
npar=35
```

O número inicial de iterações e partículas recomendado é de 100 e 25, respetivamente.

0.4.1 Convergência

Para saber se o algoritmo necessita de mais iterações e particulas, é necessário ter ativa a figura 1 (ver secção 0.5). A partir da análise da figura 1, é possível saber se o ótimo global ainda está a melhorar nas iterações finais e se são melhorias significativas. São apresentadas a seguir respetivamente uma figura em que o algoritmo já está fixo numa solução (figura 2), e outra no qual ainda está a convergir (figura 1):

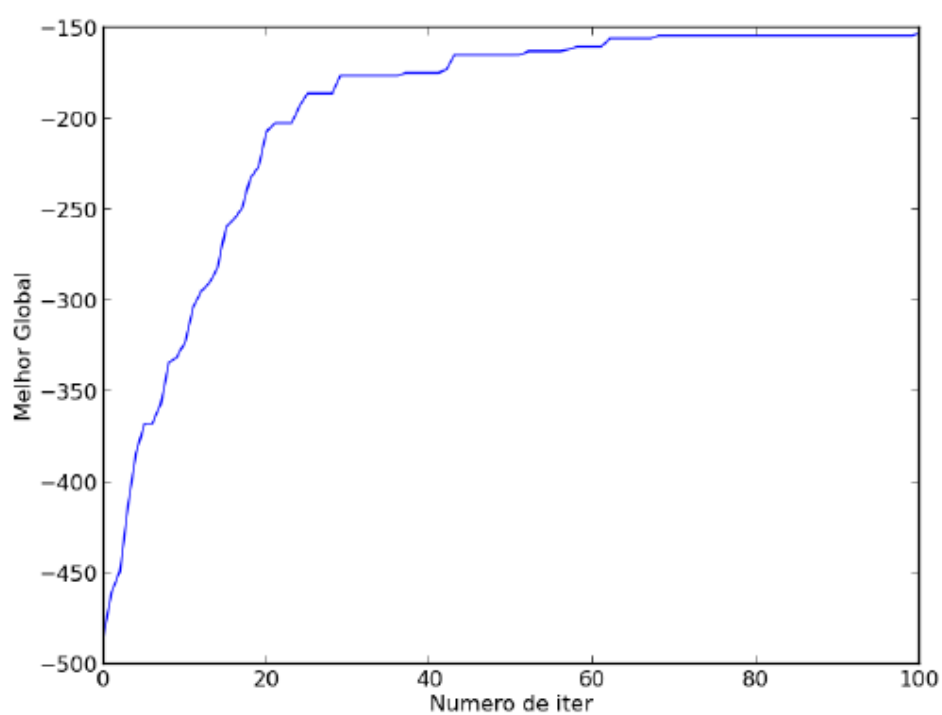


Figura 1: Caso em que o algoritmo ainda não convergiu

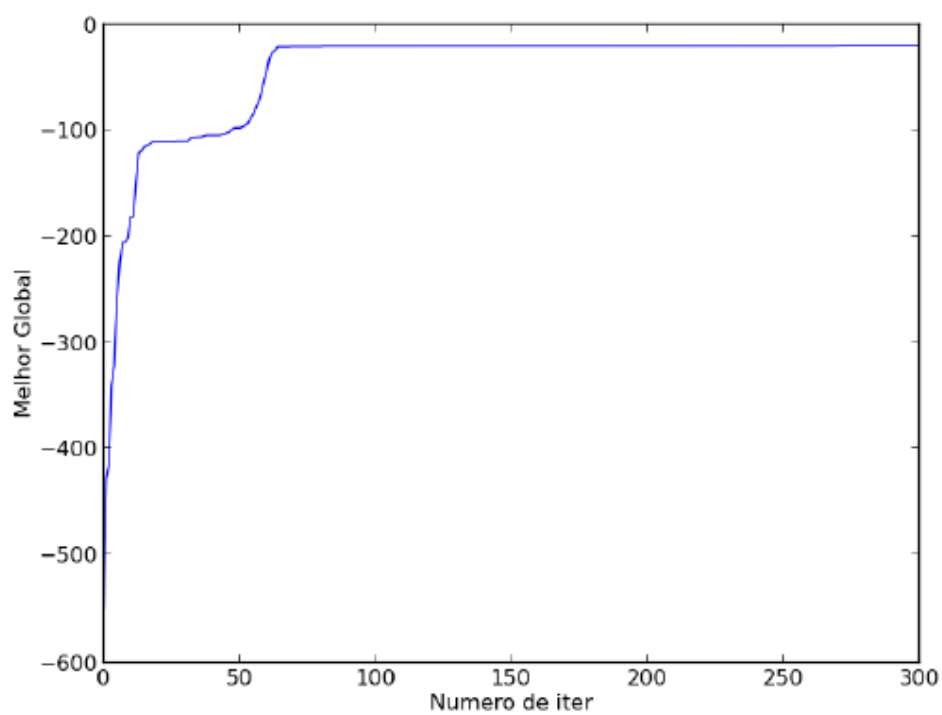


Figura 2: Caso em que o algoritmo já convergiu

0.4.2 Controlar as seeds aleatórias

Por vezes, devido às partículas iniciais do EPSO serem gerada de forma aleatória, o enxame converge para um ótimo local. Isto pode ser contornado forçando outras soluções iniciais diferentes, até se obter um ótimo global satisfatório. Esse controlo pode ser feito perto da frase chave “Sementes aleatorias”. Deve-se descomentar a linha de código que ativa as random seeds retirando o “#”:

```
#random.seed(0)
```

De seguida devem ser experimentados diferentes valores na seed, até o ótimo global ser o pretendido. Os valores devem ser inteiros. Exemplo:

```
random.seed(1)
```

0.5 Ativar e desativar gráficos auxiliares

Para além dos gráficos do módulo e da fase da resposta fornecida pelo modelo a ajustar e pelo modelo fornecido, o script calcula também um gráfico que mostra a soma das distâncias das partículas ao ponto médio do enxame, assim como um gráfico que mostra o ótimo global em cada iteração. O código que gera esse gráficos pode ser encontrado depois da frase chave “Controlo dos plots auxiliares”, e comentar e descomentar o código de cada figura, utilizando “#” antes do código. A figura 1 é o melhor global em cada iteração, e a figura 4 é a soma da distância das partículas ao ponto médio do enxame. Exemplo (figura 1 ativa, e figura 4 comentada):

```
plt.figure(1)
plt.plot(aux,fitnessbestgaux)
plt.xlabel('Numero de iter')
plt.ylabel('Melhor Global')
#plt.figure(4)
#plt.plot(aux,distancia)
#plt.xlabel('Numero de iter')
#plt.ylabel('Soma da distância das particulas ao ponto médio')
```

0.6 Ficheiros de saída

Ao terminar o algoritmo, o script grava para uma pasta as figuras do módulo e da fase do melhor global encontrado, assim como também grava os seus parâmetros num ficheiro de texto.

0.6.1 Modificar a pasta onde vão ser gravados os ficheiros de saída

Para modificar a pasta onde os ficheiros de saída serão guardados, deve procurar pelas frases chave “Ficheiros de saida 1”, “Ficheiros de saida 2” e “Ficheiros

de saída 3” e em cada colocar entre ‘plicas’ a localização de uma pasta existente e um nome de ficheiro como indicado no exemplo (note que no exemplo a localização da pasta é /home/joao/Área de Trabalho/lixo):

- Ficheiros de saída 1 (responsável por guardar o módulo da resposta do melhor global):

```
plt.savefig('/home/joao/Área de Trabalho/lixo/mod.png')
```

- Ficheiros de saída 2 (responsável por guardar a fase da resposta do melhor global):

```
plt.savefig('/home/joao/Área de Trabalho/lixo/ph.png')
```

- Ficheiros de saída 3 (responsável por guardar os parâmetros do melhor global):

```
ficheiro=open('/home/joao/Área de Trabalho/lixo/bestglobal.txt','w')
```

Note que deve ser utilizado para Ficheiros de saída 1 e 2 uma extensão do tipo “.jpg” ou “.png”, enquanto que para Ficheiros de saída 3 deve ser utilizada uma extensão do tipo “.txt”.